Automatic Crash Detection for Motor Cycles

Jussi Parviainen*, Jussi Collin*, Timo Pihlström*, Jarmo Takala*, Kari Hanski† and Aki Lumiaho†
*Department of Pervasive Computing
Tampere University of Technology, Finland
Email: firstname.lastname@tut.fi
† Ramboll Finland
Tampere, Finland
Email: firstname.lastname@ramboll.fi

***Abstract*—In this paper we present a concept for automated crash detection to motorcycles. In this concept, three different inertial measurement units are attached to head of the motorist, torso of the motorist and to the rear of the motor cycle. Crash dummy tests are done by throwing the dummy with different altitudes to simulate the effect of crash to the motorist and real data is collected by driving the motorcycle. A maximum a posteriori classifier is trained to classify the crash and normal driving. The implemented prototype system shows promising results for automatic crash detection.**

***Index Terms*—Context awareness, Pattern recognition, Wearable computers, Automotive applications, Vehicle safety.**

I. INTRODUCTION

Safety in traffic is taken into consideration more and more nowadays. Nevertheless, accidents still occur and help is usually needed as fast as possible. There has risen a discussion of automatic accident detection methods which would trigger emergency call (eCall) automatically. An example of this is a European eCall initiative[1] where in case of a crash, a vehicle automatically calls to the nearest emergency center and sends some data including exact crash location.

In order to trigger eCall, the system needs to detect accidents. The majority of the literature discusses crash detection methods for cars, e.g. [1]–[4]. Crash detection algorithms are usually based on acceleration or crush measurement. For two wheel vehicles, motorcycles, the crash detection algorithms are not thoroughly explored. In the prior art [5]–[9] the motor cycle crash detection is done using the sensors attached to the motorcycle. However, the sensors in the motorcycle may not know the severity of the crash, unlike in the cars, in the case of crash, the motorist and motorcycle may fall away from each other. In addition to triggering eCall, there have been research over motorist safety jackets, which could inflate the airbag during the crash [10], [11].

In this paper, we propose to use additional sensors to improve crash detection and situational awareness. These wearable sensors [12]–[14] are integrated to the motorists suite. Namely, one attached in torso and other one in the helmet (Fig 1). Small wireless GPS synchronized micro-electromechanical MEMS inertial measurement units (IMU) were build in order to collect data. Data from normal motor cycle driving was collected and the crash situation was simulated by dropping the dummy from different altitudes on the floor. Maximum a posteriori (MAP) classifier was build in order to infer between the crash pulse and normal driving.

[1]http://ec.europa.eu/information society/activities/esafety/ecall/index en.htm

Fig. 1. Locations of the sensor units

The paper continues in Section II with a discussion of crash pulse detection and the presentation of our classification algorithm. Next, our measurement devices and training data with the selected features are presented in Section III. In Section IV the results from the collected data are illustrated and Section V presents real time implementation of the classification system. Finally we conclude the paper in Section VI.

II. CRASH DETECTION

It is not straightforward to detect whether the crash has occurred. A known way is to follow the speed information and inclination of the motor cycle [15] or to explore the accelerations of the motor cycle and the motorist. If conventional crash pulse detection is used, where the crash is alerted using vehicle mounted accelerometer, the severity of the crash to the motorcyclist is not necessarily known. For example, when motorist falls from the motorcycle during the crash, only the motorcycle or the motorist may undergo high energy crash. Thus, it would be feasible to distribute the sensors between the motorcycle and the motorist. One option for the sensor locations was given in Fig 1. If the high crash pulse was occurred in the helmet sensor, the probability of the more severe accident is higher. This could help authorities to put more priority to more severe accidents.

In our work we take the pattern classification approach and aim to recognize the following events:

978-1-4799-4032-5/14/$31.00 ©2014 IEEE

- **Static** Motorist is not moving
- **Move** The motorist is driving a motorcycle or e.g. walking
- **zeroG** The motorist/unit is in free fall (this might happen before crash). A long free fall may implicate severe crash.
- **Peak** There is high peak in the acceleration data, this usually means crash.

In order to detect when the crash has occurred we build an MAP classifier.

A. Classification

For the classification task, we use a supervised Bayesian maximum a posteriori classifier (MAP). A supervised classification process starts by collecting a training data set with known states. This set is used to obtain

$$\mathbf{z}_j \sim N(\boldsymbol{\mu}_j, \Sigma_j), \tag{1}$$

the distribution of the observed $q-by-1$-vector $\mathbf{z}$ given that the observation comes from the class j. At this stage, the mean vector $\boldsymbol{\mu}_j(q \times 1)$ and the covariance matrix $\Sigma_j(q \times q)$ are here assumed to be perfectly known (learned from a training set). We also assume that for all classes $\Sigma_j > 0$ (i.e. Σ_j is positive definite). If (1) holds, the density function of $\mathbf{z}_j$ is

$$f_{\mathbf{z}_j}(\mathbf{z}; \boldsymbol{\mu}_j, \Sigma_j) = \frac{1}{(2\pi)^{q/2}\sqrt{|\Sigma_j|}} \exp[-\frac{(\mathbf{z}-\boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{z}-\boldsymbol{\mu}_j)}{2}]. \tag{2}$$

A new observed $\mathbf{z} = \mathbf{z}_x$ can then be classified by maximizing (2) over all the classes $j = 1...p$. To assign a probability to the classification result, we need to use unconditional prior probabilities $P(C = j)$, and assume that all the possible classes are included. Then the probabilities for all the classes are obtained from the Bayes' rule:

$$P(C = j|\mathbf{z}_x) = \frac{f_{\mathbf{z}_j}(\mathbf{z}_x) P(C = j)}{f_{\mathbf{z}}(\mathbf{z}_x)}, \tag{3}$$

where $f_{\mathbf{z}}$ is the unconditional density function for the observation $\mathbf{z}$. Prior probability can be used to control the false alarms or false negatives.

III. Measurement Device

In order to log and process the data a small wireless GPS synchronized micro-electromechanical MEMS inertial measurement units (IMU) were build. The device was designed for use as a wearable element and therefore, small size and wireless operation is required. Also, the unit should have enough processing power so it can operate the functions of an inertial measurement unit and user-defined algorithms. The device is based on an ARM Cortex M3 microcontroller [2] operating at 72 MHz with 512 kB of program memory and 64 Kb of data memory. It provides USB and micro SD card connectivity for data transfer and logging capabilities.

[2]http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1031/LN1565/PF164485

The combined accelerometer and magnetometer is ST Microelectronics LSM303DLHC2 [3] which operates over a wide acceleration sensitivity range of +- 16 g. The sensor outputs measurements with over 1 kHz rate which is necessary in highly dynamic crash situations. The gyroscope is ST Microelectronics L3GD20 [4] with user-defined sensitivity range from 250 dps to 2000 dps. Both of the sensors are connected to the microcontroller with an I^2C bus. Also, they are readily available, inexpensive, and have a small footprint.

Unit contains an u-blox UC530 GPS module [5] whose main purpose is to provide time synchronization for sensor measurements, but it can also be used to track the unit position and speed. The other radio module is Microchip RN41 Bluetooth module [6] which provides wireless connectivity. The Bluetooth protocol was chosen because of its wide use in cellular phones and tablets.

All of the aforementioned functionality is implemented on a single four-layered PCB and packaged inside a 50 mm x 50 mm x 20 mm plastic enclosure alongside a 400 mAh lithium-polymer battery.

A. Training data Collection

Normal motor cycle driving in traffic including cobblestone road was collected for null hypothesis (approximately 11 minutes). It is costly to arrange real motor cycle crash tests, however, crash pulses were simulated by throwing the crash dummy from two different altitudes 1.7 m and 3 m. Details of the crash dummy tests are given in the Table I. It should be noted that the lateral speed of the dummy was very small during throwing tests. In real life crashes speed would be greatly larger and the forces sensed by the sensors are naturally larger. However, the main difference is in the size of the crash pulse (norm of the acceleration during the crash) and the greater pulse is usually easier to recognize.

In addition, data collection was performed in a motocross track. Data included high jumps as illustrated in Fig. 2. It should be noted that the driving environment is very different in motocross track compared to normal driving in traffic. For example, if the same algorithm is used in both cases, false crash detections may occur more often in motocross track.

B. Accelerometer features

Before we can use the classification algorithm as it was described in Section II-A, the feature extraction should be done. It is known that excessive number of features can be

[3]http://www.st.com/web/catalog/sense_power/FM89/SC1449/PF251940
[4]http://www.st.com/web/catalog/sense_power/FM89/SC1288/PF252443
[5]http://www.u-blox.com/en/positioning-antennas/gnss-antenna-modules/uc530.html
[6]http://www.rovingnetworks.com/products/RN41

TABLE I. Crash dummy test details

Measurement 1	Dummy is dropped from 1,70 meters
Measurement 2	Dummy falls on his back
Measurement 3	Dummy falls frontally
Measurement 4	Dummy is dropped from 1,70 meters
Measurement 5	Dummy is dropped from 3 meters (front first)
Measurement 6	Dummy is dropped from 3 meters (back first)

Fig. 2. Jump made in motocross track

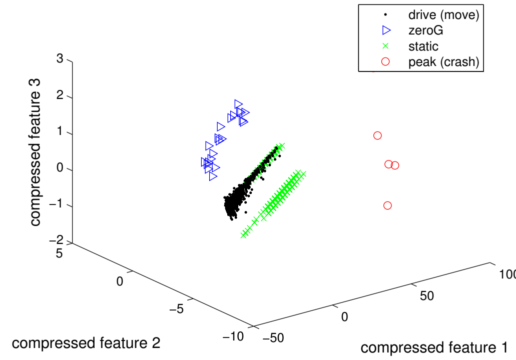

Fig. 3. Training features in the compressed domain. Each compressed feature is linear combination of original features.

extracted from accelerometer data [16] and those are also very feasibly for activity recognition [17], [18]. In this work we also concentrate on accelerometer feature based recognition.

If data from each axis of 3D accelerometer is used separately we can get information about the attitude and the inclination of the motor cycle [15]. However, it is possible to infer the states given in Section II only by using the Euclidean norm of the three dimensional accelerometer signal. The following features (Fn) are extracted from the accelerometer signal norm:

- F1: Variance
- F1: Maximum value
- F3: Mean
- F4: The number of zero crossings

In addition if we want to improve especially the detection accuracy of free fall state, we can also add features which are calculated from the full 3D accelerometer data. For example, we derived empirically two features that can be used for *zeroG* detection even if the motor is running:

- F5: The absolute value of mean of mean of each accelerometer axis

$$\mathrm{F5} = \left| \frac{\frac{1}{n}\sum_{i=1}^{n} a_i^x + \frac{1}{n}\sum_{i=1}^{n} a_i^y + \frac{1}{n}\sum_{i=1}^{n} a_i^z}{3} \right| \quad (4)$$

- F6: The variance of mean of each accelerometer axis

$$\mathrm{F6} = \mathrm{var}\left(\left[\frac{1}{n}\sum_{i=1}^{n} a_i^x \quad \frac{1}{n}\sum_{i=1}^{n} a_i^y \quad \frac{1}{n}\sum_{i=1}^{n} a_i^z \right] \right) \quad (5)$$

Here the symbols a_i^x, a_i^y and a_i^z are the ith components of x, y and z direction accelerometer readings, respectively, where the total number of samples in sliding window is n and var$(\cdot)$ is variance operator. In this work we chose some what heuristically the length of the sliding window to be around 50 ms. It should be noted that features F5 and F6 are orientation depended. Thus, these features should be trained in many different orientations. One possible approach is to multiply the data with direction cosine matrices (DCM) having different orientations and make replicas of the training data having different orientations.

As we are using data from three different units it is possible to combine features as a large feature vector (FV) from all devices

$$\mathrm{FV} = [\mathrm{F1}_{\mathrm{unit1}} \ldots \mathrm{F6}_{\mathrm{unit1}}\, \mathrm{F1}_{\mathrm{unit2}} \ldots \mathrm{F6}_{\mathrm{unit2}}\, \mathrm{F1}_{\mathrm{unit3}} \ldots \mathrm{F6}_{\mathrm{unit3}}] \quad (6)$$

and then train a classifier. Another option is to train all three units separately and then combine the output of classifiers.

In addition to classifier presented in Section II-A, it would be feasible to use time filtering, for example Hidden Markov models, as it was done in [2] for the vehicle crash detection.

IV. Results

In order to visualize the data more clearly the features compression method based on Karhunen-Loève transformation given in [19, pp. 331-334] was implemented. The method is highly similar to linear discriminant analysis (LDA), which is widely used in feature compression. After the transformation the number of features is reduced to $C-1$, where C is the number of classes. These transformed features are linear combinations of the original features. Three best linear combinations of training data features are shown in Fig. 3 which were produced after this compression method. These features were then used to train the classifier as it was described in Section II-A.

Fig. 4 illustrates the states of one crash dummy test with the classified states (above) and norm of the acceleration (below). Classifier can successfully find the all the states during the dummy test. First the dummy is in *move* state as the dummy was hold by two persons. Next in the data we can see *zeroG* state where the dummy is in the free fall and after this the *peak* state implicates the crash on the floor. Finally dummy immobile in *static* state.

The accelerometer data in the motorcycle tests was just used to find parameters for the normal driving phase when the crash peaks should not occur. As was mentioned the real motorcycle crash situations were not performed. Fig. 5 shows

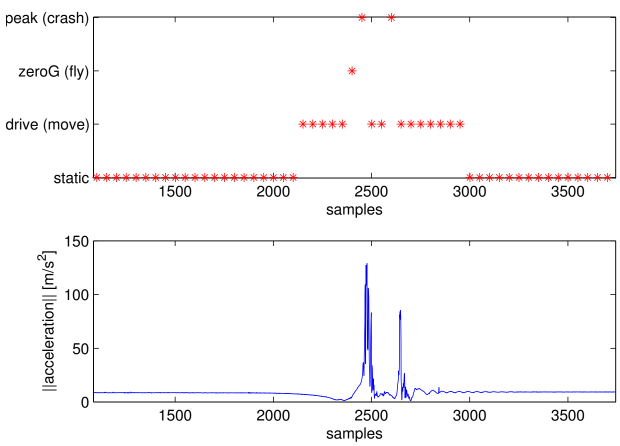

Fig. 4. States in a dummy crash test

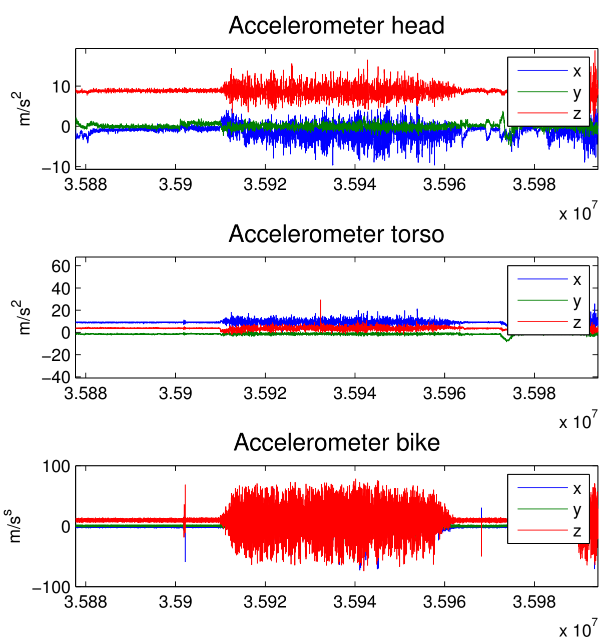

Fig. 5. 3D Accelerometer data from three different locations during motorcycle drive on cobblestone road

the raw data the cobblestone road drive. In the beginning of the figure motorcycle was stopped in the traffic lights, after that it continued driving where we can see great increase in variance. We can see directly from the figure that the different locations of the accelerometer senses different amount of vibrations during driving in the bumpy road. Thus, right amount of the training data from the each location is needed in order to get good training parameters for the classifiers; false crash alarms should not occur in any normal driving environments.

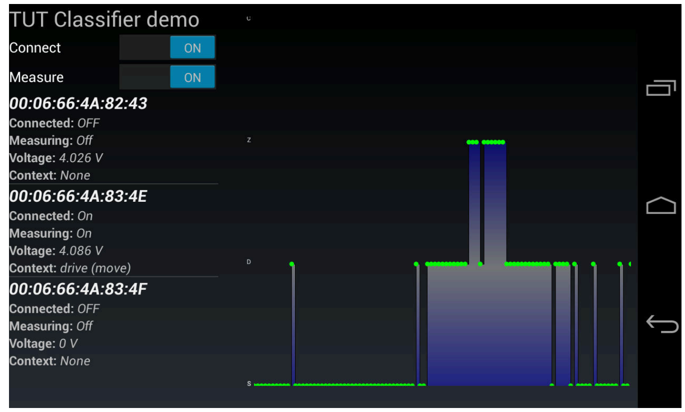

Fig. 6. Classifier results shown on the Android phone screen

V. Real Time Implemation

The feature extraction and MAP classifier was implemented on the measurement device with the help of FreeRTOS [7] tasks and queues. Two of the tasks are responsible for the operations of an inertial measurement unit. Namely, reading the sensors, calibrating the measurements and adding them into a buffer structure. The inter-task and outside world communications are based on a message system with one central communications task. All inter-task messaging and movement of the sensor data is done using thread-safe queues.

The classifier algorithm has its own task where it collects 50 accelerometer measurements into a buffer. The inertial measurement unit tasks provide these readings at a 1 kHz rate. In addition, the result of the algorithm is sent over Bluetooth using the communications task functions.

For the visualization of the algorithm data and automatic detection of a selected states an Android phone is used. The application opens a Bluetooth connection to each of the measurement units and sends a *Start measurements* message. Afterwards, the algorithm tasks of each device start receiving accelerometer measurements and send the results to the Android phone. The results are shown in a graph on the phone screen, this is illustrated in Fig. 6. Each green dot in the screen means one classification result which are updated every 50 ms. In the figure it is shown a case when the device was taken into the hand was then thrown up and caught back to hand. States are in same order as in 4 above plot.

VI. Conclusion

In this paper we presented a new concept for automated method for crash detection to motorcycles. In contrast to conventional systems, where the sensors are attached to the motor cycle, we propose a system with three different inertial measurement units were attached to head, torso of the motorist and to the rear of the motor cycle. Crash dummy tests were done by throwing the dummy with different altitude to simulate the effect of crash to the motorist and real data is collected by driving the motorcycle. A MAP classifier was trained to make decision between the crash and normal driving. When the crash occurs automated system can send automated emergency call

[7] http://www.freertos.org

and the position of the motorist. In addition to these, data containing information of the severity of the incident could be sent.

In order to really validate our classifier the real data with motor cycle crashes would be needed. However, these preliminary results show that the classifier could be applicable in real situations. In future work we will investigate how to optimally combine data from several spatially distributed inertial measurement units.

REFERENCES

[1] G. Singh and H. Song, "Comparison of hidden markov models and support vector machines for vehicle crash detection," in *Proc. Methods and Models in Computer Science (ICM2CS)*, 2010, pp. 1–6.

[2] ——, "Using hidden markov models in vehicular crash detection," *Vehicular Technology*, vol. 58, no. 3, pp. 1119–1128, 2009.

[3] C. Pinart, J. C. Calvo, L. Nicholson, and J. A. Villaverde, "Ecall-compliant early crash notification service for portable and nomadic devices," in *Proc. Vehicular Technology Conference*, 2009, pp. 1–5.

[4] C.-Y. Chan, "On the detection of vehicular crashes-system characteristics and architecture," *Vehicular Technology*, vol. 51, no. 1, pp. 180–193, 2002.

[5] H. Tabata and K. Kushida, "Automatic accident reporting apparatus for two-wheeled vehicles," Apr. 29 2009, EP Patent 1,197,426.

[6] I. Barona Martínez, F. D. García, and C. Pinart Gilberga, "Portable apparatus and method for detecting and notifying of vehicle accidents," Oct. 31 2012, EP Patent 2,518,710.

[7] Y. Kobayashi, "Impact detection sensor attachment structure for motorcycle," Mar. 8 2007, US Patent App. 11/514,212.

[8] Y. Kobayashi and T. Makabe, "Crash detection method for motorcycle airbag system with sensors on the front fork," in *Proc. 23rd International Conference on the Enhanced Safety of Vehicles*, 2013.

[9] "Continental Sensorbox for Motorcycles," http://www.conti-online.com/www/automotive_de_en/themes/two_wheelers/sensorbox_en.html, accessed: 2014-05-02.

[10] Y. Kobayashi, N. Kurata, E. Shida, Y. Wada, and M. Ishiwatari, "Motorcycle passenger protecting system," Aug. 16 2011, US Patent 7,997,613.

[11] H. Kojima and S. Ogawa, "Airbag jacket activating system," Jan. 8 2013, US Patent 8,348,304.

[12] J. Farringdon, A. Moore, N. Tilbury, J. Church, and P. Biemond, "Wearable sensor badge and sensor jacket for context awareness," in *Symp. Wearable Computers*, 1999, pp. 107–113.

[13] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *Pervasive Computing*, vol. 1, no. 3, pp. 24–32, 2002.

[14] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.

[15] H. Tabata and K. Kushida, "Automatic accident informing apparatus for two-wheel vehicle," Jul. 1 2003, US Patent 6,587,042.

[16] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[17] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proc. Wearable and Implantable Body Sensor Networks*, 2006, pp. 103–106.

[18] M. Susi, D. Borio, and G. Lachapelle, "Accelerometer signal features and classification algorithms for positioning applications," in *Proc. International Technical Meeting of The Institute of Navigation*, 2011, pp. 158–169.

[19] A. Webb, *Statistical Pattern Recognition*, 2nd ed. John Wiley & Sons, LTD, 2002.