

Smart Mobile Phone Groups-Based Design and Implementation of Vehicle Accident Self-rescue System

Long Zhao^{1*}, Jianwei You¹ and Nianmin Yao²

¹Department of Electrical Engineering and Computer Science, Foreign Language Department, East University of Heilongjiang, Harbin, Heilongjiang 150086, China

²College of Computer Science, Harbin Engineering University, Harbin, Heilongjiang 150001, China

ABSTRACT Considering the deficiency that the existing automobile accident self-rescue system requires to assemble the specific sensor, GPS and communication module, the author designed a vehicle collision or crash detection and accident self-rescue system based on the smart mobile phone groups. Taking the acceleration signals of physical moments from the smart mobile phones in the accidents car as the input samples and the assembly of acceleration data as the calculating thresholds, the system can not only improve the accuracy rating of the judgments on the accidents, but can also reduce the unusable distress signals due to the damages to individual vehicle-borne device or mobile phone in severe accidents. When the signals exceed the threshold value, the system can draw support from the functions of video, GPS positioning and 3G networking in the mobile phones automatically to call for help from the rescue center. Meanwhile, a prototype system has been implemented on Android platform. The results of the experiments indicate that the system possesses the advantages of high accuracy of collision or crash recognition, lower energy consumption and cost.

KEYWORDS

Collision
Crash
Smart phone
Smart phone groups
Three-axis acceleration sensor

INTRODUCTION

China is confronting the same troubles of highly frequent accidents as the developed European countries and the America since the time when cars are popularized.

We must acquire specific information about the venue and the number of passengers of the accident vehicles and about how they are damaged so as to minimize the loss of human and property. However, prompt rescue can't be obtained when all victims die or are injured so seriously that they can't ask for help and offer accurate information about the accidents.

Hence, it is of great practical significance to research on calling

for help automatically after car accidents. It has been realized to report accidents via specific hardware in the car in European countries. If a car is damaged seriously in the traffic accident, the hardware can dial EU's emergency call 112 to report its place to the nearest first-aid station. Such hardwares have single function and the quality varies from different manufacturers and models. In China, car accident rescue work is still based on human's calling for help through phones with only few kinds of vehicle-mounted self-help devices developed by certain factories. The devices are usual integration of sensor components, GPRS communication and GPS orientation modules. They are costly and not stable [1] and are easily broken because of being installed in the front of vehicles.

According to an authority report, China will replace America to become the biggest market for smart phones in the world in 2012. Smart phones are fast spread, which lays solid foundation for the application and development of mobile intelligent terminal. When cars crash, strong and unusual acceleration signals are generated. One built-in sensor in cell phone can record the changes of acceleration speed and do calculations on certain collision and fall

Copyright © 2016 Long Zhao *et al.*

doi: 10.18686/esta.v3i1.1

This is an open-access article distributed under the terms of the Creative Commons Attribution Unported License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

*Corresponding author: Department of Electrical Engineering and Computer Science, Foreign Language Department, East University of Heilongjiang, Harbin, Heilongjiang 150086, China. E-mail: 83993286@qq.com

detection algorithm. If a rescue is needed based on the result of calculation, it will report to the first-aid center promptly through GPS location and other functions.

ACCELERATION DETECTION

Detecting the acceleration is to judge if a car is crashed or not. The basic data needed are accumulated through a built-in triaxial acceleration sensor in a whole smart phone which is covered in monitoring network. In recent years, there have been researches on smart phone triaxial acceleration sensor. Such as in the second reference book, individual parameter and time partition and others are adopted. However, there are shortcomings of only using one single smart phone to judge if some crash has happened: (1) Smart phone can't be fixed somewhere in the car like other mobile units, so misjudgments will probably occur only depending on the acceleration signal sent by a single smart phone which its self will do irregular movement. (2) If some devices are needed to instead of human asking for help in traffic accident, it is because the accident is serious and people inside are dying or badly injured so they can not call for help. Since the car is seriously damaged in such circumstances, the probability of completeness and usability for certain device (ig. a cell phone) can't be high. If the phone is damaged, there's less possibility to call for help automatically, thus the self-rescue system is less reliable. (3) There's not accurate detection for vehicle detection. Because of the limitation of the designed systematic framework in the second reference book, a single phone can't tell if the accelerated speed "g" is that when the phone itself falls straight downward or that when the car crashes.

In this system, multi-smart phones are adopted to input the acceleration signal. The phones have free combinations and are connected into a monitoring network. When a car crash happens, all smart phones in the network can send information of accelerated speed, on-spot photos and their judgments to the server. After analyzing the information sent by phones comprehensively, the server offers rescue plans [2].

The mathematic model of the triaxial acceleration speed inside smart phone

Triaxial acceleration sensor is installed in some of the smart phones. See Figure 1, if Z axis is right above the phone screen, Y axis is where the width of screen points, and X axis is where the length of screen points; , and stand for separately the acceleration speed of the three dimensions, $|A_x|$, $|A_y|$ and $|A_z|$ are their magnitudes; if is the general acceleration, there's the formula (1):

$$|A_{sum}| = \sqrt{|A_x|^2 + |A_y|^2 + |A_z|^2} \quad (1)$$

If there's an included angle between the X, Y, Z axes of phone screen and the direction where the car is driving, its orientation is indicated by θ_x , θ_y and θ_z ; if the acceleration of Z axis projecting onto the driving direction is A_v , there's the formula (2):

$$|A_v| = |A_x \sin \theta_x + A_y \sin \theta_y - A_z \cos \theta_z| \quad (2)$$

Collision detecting algorithm

Imagine there are as many as M phones connected with detecting network, when the car has an anomaly, N out of them ($N \leq M$) send information about the anomaly and acceleration to the monitoring center. If the judgment information is accident, they send T; if not, they send F.



Figure 1 A triaxial Acceleration Model.

Through routine experiments, we know that common motions when a car is driving normally make the driver's body bend forward or backward or shake from side to side. Though there's also an acceleration, it only affects the driver's comfort with the maximum acceleration being less than 1g (g stands for gravitational acceleration). However, during a crash, the acceleration can reach instantaneously more than ten g, dozens of g, or even hundreds of g [3].

Rules are concluded out of reference and system testing:

Rule 1: When there's one phone, if $|A_{sum}| \geq 10g$, an accident is considered to happen.

Rule 2: When there's multi-phone, if M phones are connected with detecting network, when the car has an anomaly, N phones ($N \leq M$) send acceleration information A_i ($1 < i \leq N$) to the monitoring center. If

$$N=M, \text{ and } \sum_{i=2}^N |A_i| \geq M*10g, \text{ an accident occurs.}$$

When $\sum_{i=2}^N |A_i| < N*10g$, we use Rule 3 to test every phone to see whether a crash happens or not. If AXT and INT are threshold values between the maximum and minimum of $|A_{sum}|$ in a certain period, AXV and INV threshold values are between the maximum and minimum of $|A_v|$ in a certain period. Two Boolean variables, ST and SV are used to judge a crash, if it's "True", a crash occurs.

Rule 3: Judgment is made by the comparison between the extremum difference of $|A_{sum}|$ and $|A_v|$ in time quantum t1 and follow-up t2 with threshold value: if in the time quantum t1, the difference of the maximum and minimum of $|A_{sum}|$ exceeds AXT, but is less than INT in t2, ST marks "True". In the same way, $|A_v|$ is used to judge SV. When both ST and SV are True, a collision is decided to occur. The pseudocode of the calculation is as follows:

```

ST=SV=Shock=False
If  $|A_{sum1} - A_{sum2}| > AXT$  and  $|A_{sum2} - A_{sum3}| < INT$ 
ST=True
If  $|A_{v1} - A_{v2}| > AXV$  and  $|A_{v2} - A_{v3}| < INV$ 
SV=True
If ST=SV=True
Shock=True
Else Shock=False

```

All the smart phones in the car included in the detecting network send their judgment "Shock" to the server, and the server will decide according to Rule 4 whether an accident occurs or not and on rescue action in response.

Rule 4: If there're M ($M \geq N$) smart phones in the car connecting with the detecting network, and N ($0 \leq N \leq M$) of them send the

judgment “Shock” to the server, when $N > 0$, $M/N < 2$ and “Shock” value equals with the number ($> N/2$) of the phones giving True, it is decided to have a traffic accident. At the same time, the server gives orders to the smart phones connected with detecting network, and decides rescue plan by analyzing on-spot photos.

Car fall detecting algorithm

Rule 5 can be adopted to judge if a car falls in an accident. Rule 5: Under the circumstances of harmless interference, if there're M smart phones in the car connected with the detecting network, $N (0 \leq N \leq M)$ of them send to the server the vertical downward acceleration “g”, and T_1 is the time when the first phone sends “g” and T_2 is the one when last phone does, $N=0$, then there's no car fall; $N=1$, then the server wants more information of straight downward acceleration from other phones connected within the detecting network. If it is 0 or close to 0, then there's no car fall either; if it is close to “g”, other means can help to judge for it is also likely to have a fall incident. If $1 < N \leq M$, and $T_2 - T_1 < 1$ s, when $N=M$, a car fall occurs; when $M/2 < N < M$, it also occurs; when $N < M/2$, the server wants more information of straight downward acceleration from other phones connected within the detecting network with the help of other judging means [4].

SYSTEM IMPLEMENTATION AND TEST

A set of prototype system based on Android client is established to test the practicability of detecting algorithm. The development platform is a brand installed Windows XP, two MOTOROLA XT300 and one HTC One V with Java and Android-sdk-windows.

Systematic phones and implementation

With C/S being the framework, the system combines smart phone client and server. The client analyzes the condition through gathering the changes by triaxial acceleration sensor and judges the occurrence of accidents and sends its judgments, and acceleration and on-spot information if necessary to the server. The server sorts out, analyzes and judges the comprehensive information sent by all smart phones in the car connected in the detecting network. The system server combines Web server and file server. The Web adopts Apache Tomcat 6.0 server, and the file server adopts Microsoft SQL Server 2005 database to manage files. The client side is connected with the server through 3G, GPRS, WiFi etc.

System's main operational process is as Figure 2.

In Android system, it is not complex to carry out the application based on acceleration sensor, because the Android system provides strong management for the sensor. In fact, the managements to all types of the sensors are the same, the differences only lie in the types' differences of the sensors.

The application of the sensor goes like following procedures:

1) Sensor Manager object stands for the management on sensors of the system, through calling the Context and get System Service (Context.SENSOR_SERVICE) to get Sensor Manager object.

2) Through the get Default Sensor (int type) of Sensor Manager to get appointed typed sensor.

3) To call register Listener from Sensor Manager in the on Resume of the activity method to register and monitor the appointed sensor. Through monitoring, data is acquired sent back from sensor. There're two ways to realize Sensor Listener interface.

(a) Void on Accuracy Changed (int sensor, int accuracy): the way to be used when the accuracy of sensor changes.

(b) Void on Sensor Changed (int sensor, float values): the way to be used when the data in the sensor changes. Major service code to develop sensor application should be carried out here, for

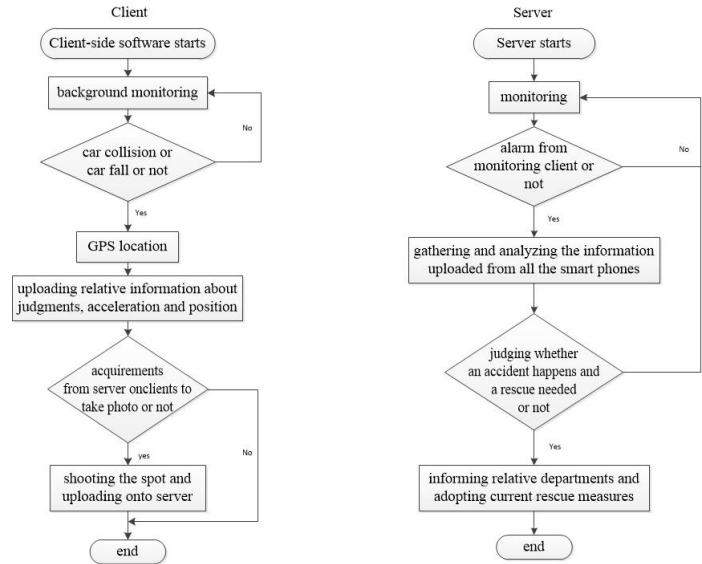


Figure 2 System Operational Process.

example, to read data and operate according to the variation etc. The input parameter “sensor” is a constant represents sensor type; its values is of the type of “float” array, whose length and details vary with different types of sensors.

System test

Compared with PC client, the performance of mobile client is not good enough. Besides, its stand-by time is limited. Hence, the energy consumption measurement is needed to test the client-side software [5,6]. It includes stand-by time variation along with the operation of the software and memory usage. The basic frequency is 550Hz, and the internal storage is 256MB. The Table 1 is the test result of MOTOROLA XT300 with the battery capacity of 1130mAh.

Table 1 Stand-by time and memory usage.

Operation of software installation	Stand-by time of cell phone (h)	Memory usage of cell phone (MB)
Before installation	48	80
No operation	48	82
Operation	32	115

Another important indicator needed to be test is the alarm accuracy and the probability that the equipment can be reused after severe accident. Compared single smart phone and multi-smart phones, both as system clients, through testing, the differences between above two indicators are as Table 2.

Table 2 Alarm accuracy and equipment survival probability.

Client form	Reusable probability of client	Alarm accuracy
Single	50%	78.9%
Multi-phones	98.9%	96.7%

The above two tests show that smart phones' stand-by time and performance in this system client software is acceptable. The alarm accuracy using multi-phones as system client is higher than using single phone. Besides, the probability of available client in severe traffic accidents is also higher.

CONCLUSIONS

There are many advantages to adopt multi-smart phones which can be connected to network freely in this system as the clients. First, it solves the problem of false alarm because of sudden changes of the phone's self-acceleration and other special reasons. Second, it avoids the trouble because of damaged and useless client in serious accident. Finally, it reuse phones own functions such as GPS location and takes photos to help rescue without increasing the cost of hardware. This system bears novelty design and wide utility value and bright prospect for market and business.

ACKNOWLEDGMENT

This work is supported by the Heilongjiang Provincial Education Science 'the 12th Five-Year Plan' Planned Project (No. 12543048).

REFERENCES

1. Campo, E., & Grangereau E. (2008). "Wireless Fall Sensor with GPS Location for Monitoring the Elderly". *30th Annual International IEEE EMBS Conference*, August, 498–501.
2. Li, W. J., Lin, Y. P., & Ye, S. T. (2011). "The application of Smart Phones in Collision Detection and Self Rescue in Car Accident". *Computer Engineer*, (9), 245–247.
3. Chen, Y. L., & Wang, C. A. (2008). "Vehicle Safety Distance Warning System Novel Algorithm for Vehicle Safety Distance Calculating between Moving Cars". *Proceedings of the IEEE International Conference on Industrial Technology*. Chengdu, 2570–2574.
4. Li, G. (2011). "Crazy Android Lecture". *Electronic Industry Press*.
5. Gong, Y. Z. (2008). "Software Testing Course". *China Mechine Press*.
6. Dalal, S. R., Jain, A., Karunanithi, N., & Leaton, J. M. (1999). "Model-based testing in practice". *In 21st Int. Conf. on Software Engineering*, July, 285–294.