A Pedestrian Safe Walking Detection System based on Smartphone Sensors

Fengtao Xue, Hailong Hu, Gaoliang Yan, and Yantao Li*

College of Computer and Information Sciences, Southwest University, Chongqing 400715, China

* Corresponding author e-mail: yantaoli@swu.edu.cn

Keywords: Safe Walking Detection; Smartphone Sensors; Opencv4android; Movement Characteristics of Human Eyes.

Abstract. In recent years, pedestrians using smartphones while walking for entertainment has become more and more popular. To avoid stumble, fall down, or even collide with other pedestrians, we propose *Walk Well*, a safe walking detection system that uses smartphones to provide pedestrians with more safety on watching smartphone while walking in this paper. More specifically, we first obtain the moving speed of the pedestrian using the gravity and accelerometer sensors on smartphones. Then, we exploit the front camera based on OpenCV4Android to detect pedestrian's face and eyes, and if the watching time reaches a threshold we set, *Walk Well* will give a vibration alert to pedestrian through the smartphone. Finally, we implement *Walk Well* on an Android smartphone, and evaluate the accuracy of the performance, and the results show that it can prevent the potential danger for pedestrians staring at smartphones.

1. Introduction

Smartphones are no longer exclusively used as telephones, but mainly as different kinds of applications. For example, nowadays, people pay more attention to the multimedia entertainment of smartphones instead of only phone call. When pedestrians are reading news, sending messages or playing mobile games, in these cases, they have to focus on the screens of their smartphones rather than the surroundings. It has been shown that pedestrians using smartphones to do other activities may lead to inattentive blindness, which is more likely to cause serious accidents [1]. There could be some injuries happening on pedestrians who are playing with their smartphones, and some funny situations can also happen on other passersby as well, when pedestrians stumble, fall down, or even collide with someone who is also focusing only on the smartphone's screen.

In order to solve this problem, some existing related works have been studied. The authors in [2] used the smartphone's camera to detect vehicles moving to the pedestrian, and alert the pedestrian of a potentially unsafe situation. They utilized machine learning algorithms to distinguish cars from empty roads and tried to warn the user when the danger is impending. In order to spot and avoid obstacles, the authors in [3] utilized an array of custom-built ultrasound sensors combined with a vibration jacket and an ARM9 based embedded system to assist visually impaired people. However, these approaches need users to take special hardware with them than they normally would not carry along, requiring extra effort and monetary investment.

Considering that the extra hardware is expensive and inconvenient, in this paper, we introduce a smartphone-based safe walking detection system, *WalkWell*, using solely an Android smartphone to detect the behavior of pedestrians. More specifically, we first provide an algorithm which estimates the moving speed of the pedestrian using the gravity and accelerometer sensors on smartphones. Then, if the walking speed of pedestrians reaches the threshold of a normal walking speed, we activate the front camera based on OpenCV4Android [4] to detect the pedestrian whether he is watching the smartphone's screen or not. Finally, if the watching time period reaches the threshold of a normal watching time, *WalkWell* will give an alert via vibrations to the pedestrian. In addition, taking the battery consumption into consideration, the detection system is designed to run in the background. Finally, we implement *WalkWell* on an Android smartphone and evaluate the accuracy of the performance, and the results show that it can prevent the potential danger for pedestrians staring at smartphones.

2. System Design

In this section, we describe the system architecture of *WalkWell* in detail, as illustrated in Fig. 1. As shown in Fig. 1, *WalkWell* is composed of two modules: 1) Walking Detection, and 2) Face and Eyes Detection.

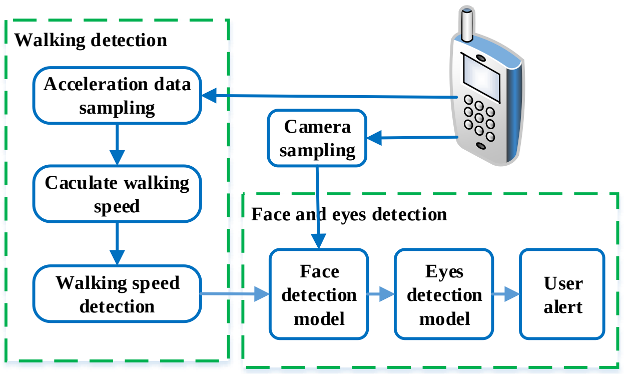


Figure 1. The detailed system architecture of WalkWell.

2.1 Walking Detection

As we known, the accelerometer of the android system has three components along x-axis, y-axis and z-axis, which can produce three acceleration a_x, a_y and a_z[5]. Therefore, we define a phone coordinate system, as illustrated in Fig. 2(a), where the X-axis is horizontal pointing to the right, the Y-axis is vertical pointing upward, and the Z-axis proceeds from the front to the back of the screen.

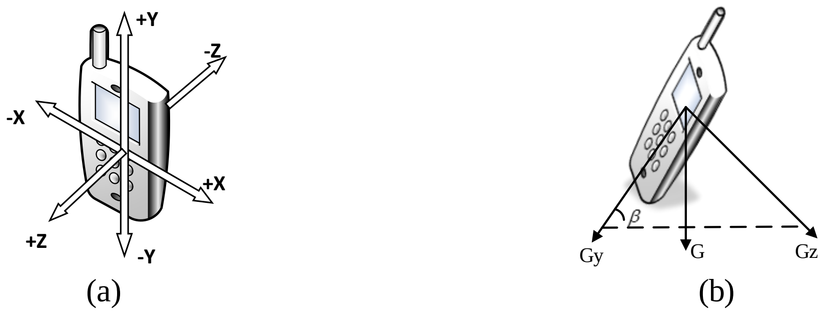


Figure 2. Phone coordinate system.

When the pedestrian holds a smartphone, there is an angle β between smartphone's y-axis and walking direction, as shown in Fig. 2(b). We can use the angle β and the acceleration of y-axis a_y to calculate the walking speed, because no matter the smartphone user is in walking or stationary state, the smartphone's gravity always exists. Consequently, we can utilize the component of gravity to calculate the angleβ, as shown in Eq. (1):

$$\tan\beta = \frac{G_y}{G_z}, \beta = \tan^{-1}(\frac{G_y}{G_z}) \tag{1}$$

As illustrated in Fig. 2(b), G_y and G_z are the components of gravity. When the detected pedestrian is walking, the acceleration of y-axis is a_y obtained from the accelerometer of the smartphone. The walking acceleration $a_{walking}$ and $v_{walking}$ are obtained from Eq. (2):

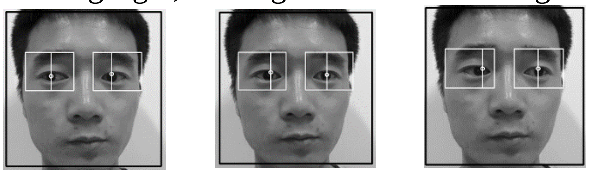
$$a_{walking} = a_y \cos\beta, \; V_{walking} = \int_{t_0}^{t_1} a_{walking} \tag{2}$$

Where $a_{walking}$ is the acceleration data sampled at timet_0, T is time difference from t_0 and t_1 which is a very short time interval, and $v_{walking}$ is calculated by integration of $a_{walking}$ with the corresponding time interval.

2.2 Face and Eyes Detection

We introduce face and eyes detection in this section. First, we describe the process of the face and eyes detection which is developed using AdaBoost algorithm and OpenCV4Android library. Then, we use the grayscale image detection algorithm to determine the accurate eyes movement mode of the detected pedestrian.

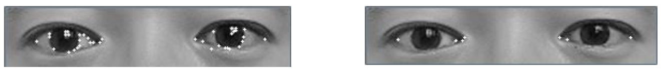
OpenCV4Android is available as a SDK with a set of samples and Javadoc documentation for OpenCV Java API. The whole processes are as follows: 1) loading the cascade classifier file from native resource file, 2) detecting face and eyes using front camera of the phone, and 3) releasing the cascade classifier file. The result of detection is showed in Fig.3. As depicted, Figs. 3(a), 3(b) and 3(c) are showed as the eyes turning right, looking forward and turning left, respectively.



(b) (c)

Figure 3. Face and eyes detection.

In general, the corner on the edge of the image has the largest curvature change rate, and the majority of corners are on the edge. Corner detection method based on the difference of local grayscale of image is mainly used in pixel point's analysis, and one of the classic algorithms is Harris corner detection [6]. We detect the corner points in the eyes picture using Harris corner detection algorithm, and the results are shown in Fig. 4. From our practical experiment, we found that the number of corner points exceeded our expectation, as shown in Fig. 4(a). According to the relative positions we searched, the corner points are close to the edge of the triangle. Hence, we select 7 points near the left and right edges of the human eye image respectively to make the system more efficient, as shown in Fig. 4(b). Then, we calculate the mean value of the number of left and right corner as the optimal eye corner point.



(a) (b)

Figure 4. Corner detection.

After we obtain the optimal eye corner, our main task is to recognize eyeball's location. We use Canny edge detection algorithm to detect the eyes' edge, modelling a closed region composed of the edge and left, right corners, as shown in Fig. 5. In Fig. 5, point *A* is the eye's right corner, while point *D* is the eye's right corner. Points *B* and *C* split Line *AD* into three equivalent parts. We define the closed region as three regions *a*, *b* and *c*, and the corresponding eyes movement modes are mode *a* (turn left), mode *b* (look forward), and mode *c* (turn right). We set a grayscale threshold *T* (*T*=60) to determine which part is the eye in which corresponds to the eyes movement mode. In the grayscale picture, the eyeball has a lower gray value, therefore, if the eyeball is inside a specific part, the number of the pixel with grayscale will be less than threshold *T*. As a consequence, our system is indending to detect the mode *a*, because it stands for the pedestrian staring at the smartphone's screen.

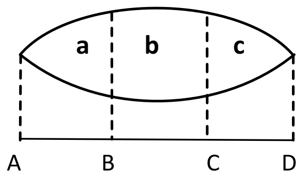


Figure 5. Eye model.

3. Performance Evaluation

In the experiments, our system was developed on an Android platform, using SDK version 18 and version 4.3 of the operating system [7]. We solely implement *WalkWell* on an Android based MX5 smartphone.

To evaluate the practical viability of our system, we conducted some real experiments. We recorded the time when people were walking while playing smartphones, with the situation that the pedestrians were staring at the smartphone's screens and walking on a straight road. We found if the time of pedestrian staring at smartphone's screen exceeded about 6 seconds, the pedestrian deviated from the original direction, hesitating to walk forward or suspending to adjust the view of sight. As a consequence, we set the threshold time as 6 seconds in our system. If the time period of pedestrian staring at the phone's screen exceeds 6 seconds, our system will give an alert via vibration to prevent the pedestrian from potentially dangerous situations. Next, we researched people's general usage behavior about the distance from eyes to the phone's screen which is about 30cm. If it is in close proximity to this distance, our system can detect the face and eyes movement accurately. We recorded the eyes' pictures of pedestrians when they were staring at the screen, and calculated the number of the pixel less than T as shown in Table 1. For Look Forward, the number of the pixel less than T in region b is more than that in regions a and c, which indicates the pedestrian is staring at the screen. Those proved our expectation. We found that if the time of a desired number of grayscale keeps in region b exceeding 6 seconds, *WalkWell* will send a vibration alert to the pedestrian. In contrast, Turn Right and Turn Left show that the pedestrian is looking at surroundings, rather than the screen, so the pedestrian is safe in this case.

TABLE 1. The number of pixel less than T.

Eyes movement	a	b	c
Turn Right	103	31	10
Look Forward	60	292	48
Turn Left	24	126	168

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant nos. 61528206 and 61402380), the Natural Science Foundation of CQ CSTC (Grant no. cstc2015jcyjA40044), and the Fundamental Research Funds for the Central Universities (Grant nos. XDJK2013B030, XDJK2015B030 and SWU111047).

References

[1]. I. E. Hyman, S. M. Boss, B. M. Wise, et al. Did you see the unicycling clown? Inattentional blindness while walking and talking on a cell phone. Applied Cognitive Psychology, 2010, 24(5): 597-607.

[2]. T. Wang, G. Cardone, A. Corradi, et al. WalkSafe: A pedestrian safety app for mobile phone users who walk and talk while crossing roads. HotMobile'12, ACM, 2012.

[3]. B. S. Shin and C. S. Lim. Obstacle detection and avoidance system for visually impaired people. HAID, 2007.

[4]. http://opencv.org/platforms/android.html.

[5]. http://www.android-doc.com/guide/topics/sensors/sensors_overview.html.

[6]. J. B. Ryu, H. H. Park, J. Park. Corner classification using Harris algorithm. Electronics Letters, 2011, 47(9):536-538.

[7]. Y. Li, F. Xue, L. Feng, Z. Qu. A driving behavior detection system based on a smartphone's built-in sensor. International Journal of Communication Systems, DOI: 10.1002/dac.3178.