

## VEHICLE DETECTION USING ANDROID SMARTPHONES

Zhiquan Ren<sup>1</sup>, Chun Wang<sup>2</sup> & Jibo He<sup>3</sup>

<sup>1</sup>Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China

<sup>2</sup>Department of Psychology, University of Minnesota, Minneapolis, Minnesota, USA

<sup>3</sup>Department of Psychology, Wichita State University, Wichita, Kansas, USA

Email: jibo.he@wichita.edu

**Summary:** Rear-end collisions are the most common traffic accidents. Technologies, such as collision warning systems, are developed to reduce the risks of rear-end collisions. This study presents a mobile technology using smartphones to detect the leading vehicle, allowing the possibility to make collision warning systems more affordable and portable. This technology uses the rear camera of an Android smartphone to capture images of driving scenes, and then uses advanced computer vision algorithms to detect and track the leading vehicle. This study may have important applications to improve driving safety.

### INTRODUCTION

A rear-end collision refers to a vehicle accident in which the front of one vehicle collides with the rear of another vehicle. It is one of the most common types of traffic accidents, causing a large number of injuries, fatalities, and financial losses every year (Lee, McGehee, Brown, & Reyes, 2002; Singh, 2003). According to data from the General Estimates System (GES) and the Fatality Analysis Reporting System (FARS) databases, rear-end collisions represent 23% of all collisions, 1.45 million out of 6.26 million police reported crashes per year (Wilson, Miller, Burns, Chase, Taylor, Butler, McGehee, & Dingus, 1998).

To reduce the risks of rear-end collisions, advanced technologies, for example, forward collision warning systems, RADAR and LIDAR, have been developed. Forward collision warning systems can provide drivers with a warning prior to a potential collision, which allows drivers to take preventative actions (Cualain, Glavin, Jones, & Denny, 2007). Studies showed that collision warning systems can help drivers avoid more collisions, regardless whether the drivers are distracted or not (for example, Lee, McGehee, Brown, & Reyes, 2002).

To reduce traffic accidents and make collision warning systems more affordable for drivers, this study proposes to use a smartphone as a collision warning system. The increasing popularity of smartphones makes a smartphone-based collision warning system very cost-effective for existing smartphone owners. As of May 2011, as much as 35 percent of American adults owned a smartphone, according to the study by the Pew Research Center (2011). A smartphone-based collision warning system will be more portable and affordable than existing collision warning systems which are built in some luxury vehicles.

The algorithms for vehicle detections based on computer vision can be classified into two major categories, that is, feature-based and model-based algorithms.

*Feature-based algorithms.* The feature-based algorithms use representative features of vehicle for detection. These vehicle features include color, symmetry, edges, shadows, and rear-lights (Sun, Bebis, & Miller, 2006). The major limitation of feature-based algorithms is that many features are not always present. For example, vehicle shadow is only present in daytime under sunlight, and not available for vehicle detection at nighttime or when the leading vehicle is under the shade of road-side trees.

*Model-based algorithms.* The model-based algorithms train a descriptor or a classifier of vehicles using vehicle databases, for example, using Neural Networks or Support Vector Machines (SVMs) (Sun, Bebis, & Miller, 2006). The model-based algorithms are usually used to verify a detected object because scanning the whole image with learning algorithms can be time consuming.

The proposed system in this study uses a Smartphone as a forward collision warning system. The forward facing video camera of a smartphone captures images of traffic scenes, then feeds the images to the CPU of the smartphone for image processing. Intel's open-source computer vision framework, the OpenCV 2.3 for the Android operating system, is used to develop the computer vision algorithms for vehicle detections.

## **METHODS**

### **Equipment and platform**

The algorithms are developed using a Samsung Galaxy Note smartphone. The smartphone runs the Android 2.3.6 operating system with the kernel 2.6.35.11. The smartphone has a CPU of 1.4 GHz dual-core ARM Cortex-A9 (GT-N7000), with 1 GB RAM memory. The battery is Li-ion 2500 mAh. The rear camera is 8 Mpx  $3264 \times 2448$  with auto focus.

The computer vision algorithms are implemented using the Java programming language and the OpenCV 2.3 computer vision framework for the Android. The OpenCV framework is developed and released under a BSD license by Intel, and freely available for both academic and commercial applications.

### **System setup**

Figure 1 shows the setup of the proposed system in real-world driving. The smartphone can be mounted either on the dashboard or the windshield of a vehicle. The smartphone is placed horizontally with the rear camera pointing towards the road ahead.



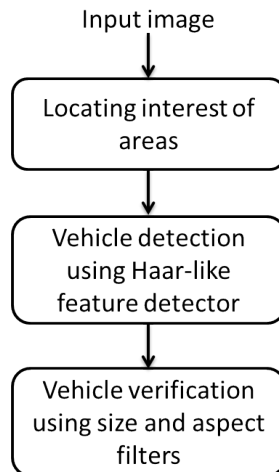
(a). A smartphone mounted on the dashboard (b). A smartphone mounted on the windshield  
**Figure 1. The system setup of vehicle detections using a smartphone**

**Algorithm framework**

The vehicle detection algorithms are carried out in three steps. The first step is locating the areas of interest. Computer vision performance depends heavily on the image size, which is especially a great concern for smartphone applications considering the limited computation resources of smartphones. The second step is identifying candidate vehicles. Vehicles are located using Haar-like feature detectors within the areas of interest. The third step is to verify candidate vehicles. The vehicles identified using the Haar-like features may be false-alarms, i.e., non-vehicles are identified as vehicles. The third step uses the size filter and the aspect ratio filter to verify the candidate vehicles. Figure 2 depicts the algorithm framework for vehicle detections.

**Image preprocessing**

The camera of the Samsung Galaxy Note smartphone captures images at the resolution of 720×1280. The raw image is first reduced to a resolution of 180 × 320, a quarter of its original resolution. Then, the resized color image is converted to a grey image. Image resize and grey/color conversion can greatly reduce the array size of images, which make the smartphone application run much faster. Lastly, the bottom half of the resized grey image is set as the area of interest to detect vehicles.



**Figure 2. The algorithm framework for vehicle detections**

## Haar-like Feature Detector

A Haar-like feature detector is used to detect candidate vehicles. The Haar-like feature using the AdaBoost algorithm is commonly used in face detections because it is fast and accurate (Viola & Jones, 2004; Viola, Jones, & Snow, 2003).

A Haar-like feature considers adjacent rectangles at a specific region in a moving detection window. An image area can be described as the combination of different Haar-like features. The number and type of Haar-like features can be used to represent different objects. If we define an accumulated sum of intensity from the origin as

$$S(i, j) = \sum_{x=0}^i \sum_{y=0}^j I(i, j) \quad (1)$$

where  $I(i, j)$  is the intensity at the location  $(i, j)$ ,  $S(i, j)$  is the accumulated sum of intensity from origin at the location  $(i, j)$ .

The sum of intensity of a rectangle in the image, defined as two points at  $(x_{left}, y_{up})$  and  $(x_{right}, y_{down})$ , can be easily calculated as the following equation, which greatly speeds up the computation.

$$S_{acc}(x_{right}, y_{down}) - S_{acc}(x_{left}, y_{down}) - S_{acc}(x_{right}, y_{up}) + S_{acc}(x_{left}, y_{up}) \quad (2)$$

Following the procedure of face detections suggested in Viola and Jones (2004), we trained a Haar-like classifier using the AdaBoost algorithm. To train the Haar-like classifier for the rear-view of vehicles, we used 520 positive images (images of vehicles), including 126 images from the “Cars 1999 (Rear) dataset” from the computer vision group of the California Institute of Technology and 194 images from “CBCL CAR DATABASE” of MIT, and 200 images we took in the city of Wichita, KS using an Samsung Galaxy Note Android smartphone. We used 520 negative images without vehicles, which were cropped from actual driving scenes. The negative images included roads, curbs, trees, buildings, traffic signs, and traffic lights.

## Size filter

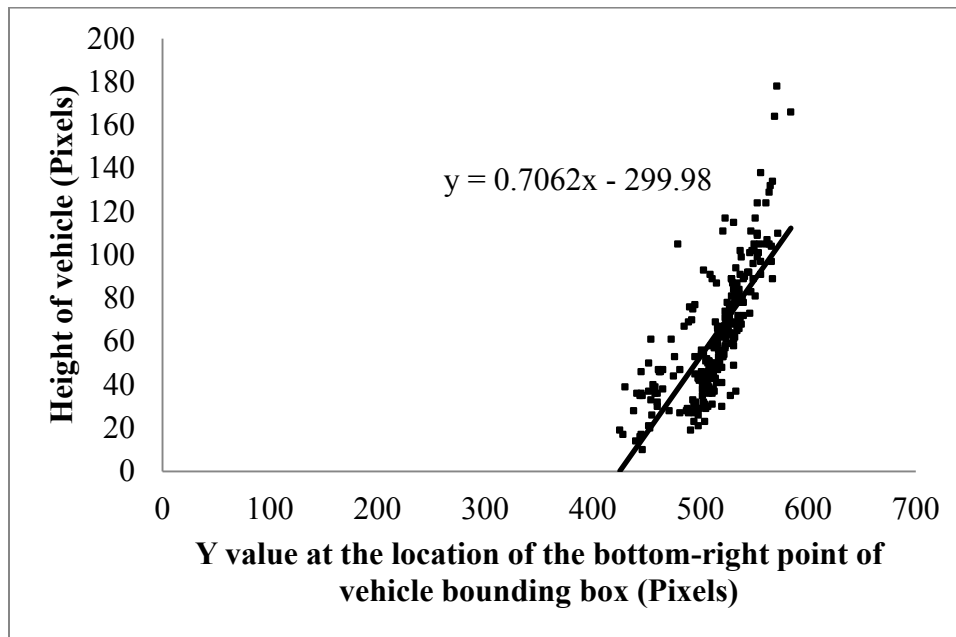
A size filter is used to verify candidate vehicles detected by the Haar-like feature. According to 3D perspective, vehicles further ahead should be smaller. In a 2D image with the origin of the coordinate system at the top left corner of the image, the height of a vehicle should be smaller if the Y value is smaller, where the Y value is the bottom-right point of the vehicle bounding box. Figure 3 depicts the height of a vehicle in pixels at different locations of in the images. This figure is based on 258 vehicles cropped from real driving scenes captured using the Samsung Galaxy Note smartphone. If the height of a vehicle detected by the Haar-like feature is too large or too small as predicted by its location in the image, this candidate vehicle is rejected, and excluded from further processing.

### Aspect ratio filter

Most vehicles have an aspect ratio (width/height) from 0.4 to 1.6 (Teoh & Bräunl, 2012). This range is used to further verify the candidate vehicles detected using the Haar-like feature detector. If the bounding box of the candidate vehicle is defined by its upper left coordinate ( $x_{left}$ ,  $y_{up}$ ) and lower right coordinate ( $x_{right}$ ,  $y_{down}$ ), the aspect ratio of the bounding box can be calculated as:

$$\text{Aspect ratio} = \frac{x_{right} - x_{left}}{y_{down} - y_{up}} \quad (3)$$

If the aspect ratio of the bounding box is within the range of 0.4 to 1.6, it is accepted as a vehicle; otherwise, the candidate vehicle is rejected.



**Figure 3. The relationship of vehicle height and vehicle location**  
(The origin of the coordinate system is at the top left corner of an image.)

### DISCUSSIONS

To reduce rear-end collisions, we developed a smartphone based technology for vehicle detections, which can be a portable and affordable alternative to the current expensive collision warning systems.

The Haar-like feature detector using the AdaBoost algorithm is used to detect candidate vehicles. We trained the Haar-like feature detector for the rear-view vehicles using 520 positive images. Further efforts are needed to expand the database of training images to make performance robust under different lighting, weather, and road conditions, and for different vehicles.

Although the Haar-like feature is a time efficient approach for vehicle detections, it also produces a number of false positives, i.e., non-vehicle regions of the images are identified as

vehicles by error. The size filter and the aspect ratio filter are used to further verify the candidate images, and reduce false positives.

With the limited computational resources of smartphones, our algorithms still achieve a high sampling rate of about 7Hz, which is fast enough for practical applications. We strive to get this high sampling rate by grey color conversion, image resize, and localization of the area of interest. This research contributes to the efforts of vehicle detections in two major aspects: first, it provides a working prototype for real-time vehicle detections using an Android smartphone; second, it offers a time-efficient approach to verify vehicle candidates, and achieve a high sampling rate for vehicle detections.

## ACKNOWLEDGMENTS

The content of this paper only reflects the personal opinions of the authors, and does not reflect the ideas of any organizations the authors have worked or are currently working for. This research is not supported or funded by the authors' affiliations.

## REFERENCES

- Cualain, D.O., Glavin, M., Jones, E., Denny, P. (2007). Distance detection systems for the automotive environment: A review, In *Irish Signals and Systems Conference*, Derry, N. Ireland, 2007.
- Lee, J. D., McGehee, D. V., Brown, T. L., & Reyes, M. L. (2002). Collision warning timing, driver distraction, and driver response to imminent rear end collisions in a high-fidelity driving simulator. *Human Factors*, 44(2), 314-334.
- Pew Research Center. (2011). 35% of American adults own a smartphone. Retrieved on Oct 22, 2012 from [http://pewinternet.org/~media/Files/Reports/2011/PIP\\_Smartphones.pdf](http://pewinternet.org/~media/Files/Reports/2011/PIP_Smartphones.pdf)
- Singh, S. (2003). Driver attributes and rear-end crash involvement propensity. NHTSA Technical Report, DOT-HS-809540.
- Sun, Z. , Bebis, G. , & Miller, R . (2006). On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (5), 694-711.
- Sun, Z. , Bebis, G. , & Miller, R . (2002). Improving the performance of on-road vehicle detection by combining Gabor and wavelet features. *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, 130-135.
- Teoh, S.S., & Bräunl, T. (2012). Symmetry-based monocular vehicle detection system. *Machine Vision and Applications*, 23:831–842.
- Viola, P., & Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57 (4), 137-154.
- Viola, P., Jones, M., & Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance, *9th IEEE International Conference on Computer Vision (ICCV 2003)*, 14-17 October 2003, Nice, France, 734-741.
- Wilson, T., Miller, S., Burns, M., Chase, C., Taylor, D., Butler, W., McGehee, D.V., & Dingus, T.A. (1998). Light vehicle forward-looking, rear-end collision warning system performance

guidelines. Department of Transportation National Highway Traffic Safety Administration  
Office of Crash Avoidance Research (NRD -51).