2013 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
AUTONOMOUS GROUND SYSTEMS (AGS) MINI-SYMPOSIUM
AUGUST 21-22, 2013 - TROY, MICHIGAN

LOW COST ADVANCDED DRIVER ASSISTANCE SYSTEM (ADAS) DEVELOPMENT

Hong Bae, Ph.D.
Manager,
Vehicle Systems
IAV Automotive Engineering, Inc
Northville, MI

Yi Jiang
Senior Software Engineer,
Vehicle Systems
IAV Automotive Engineering, Inc
Northville, MI

Chris Hennessy
Director,
Vehicle Systems
IAV Automotive Engineering, Inc
Northville, MI

ABSTRACT

Popularity of Advanced Driver Assistance Systems (ADAS) in the passenger car industry has seen an explosive growth in recent years. Some ADAS that are becoming ubiquitous are Lane Departure Warning (LDW), Blind Spot Detection (BSD) and automatic parking or parking assistance systems. In many cases, such systems had been developed specifically to handle the most demanding driving conditions at very high speeds, which typically require very sophisticated software and high-power hardware. However, in the other application areas or geographical regions, such sophistication often hinders adoption of the technology.

An alternate approach is to use off-the-shelf (OTS) component as much as possible so that similar systems with an appropriate subset of functions can be developed cheaply and quickly. The approach similar to the NASA's "PhoneSats" program is discussed in this paper.

INTRODUCTION

Popularity of advanced driver assistance systems (ADAS) has soared in recent years. Frost & Sullivan estimates that, by 2018, 40% of compact cars would have some kind of ADAS features. These features, once reserved for much more expensive luxury cars, include Adaptive Cruise Control (ACC), Forward Collision Warning (FCW) and Lane Departure Warning (LDW), Furthermore, new public policies such as EU's Transport Policy 2011-2012 which aims to reduce road fatalities by 50%, proliferation of ADAS in all spectrum of automotive application is accelerating. Also, other industries and sectors (e.g., heavy trucks, transit buses, and heavy machinery) are adopting rapidly the ADAS technologies developed for passenger car domain. This paper describes a recent R&D activity that aims to maximize synergies out of development in many different industries.

LOW COST SYSTEM DEVELOPMENT IN OTHER INDUSTRIES

Many variables contribute to the "low cost" development: development time, learning curve, proprietary hardware and software, and flexibility. Combination of many technological development has contributed to the idea in this project:

NASA PhoneSat

In 2009, NASA, under Small Spacecraft Technology Program, started building small satellites (Nano-satellites) using unmodified consumer grade smart phones. These smartphones already have a wealth of capabilities needed for satellite systems, including fast processors, versatile operating systems, multiple miniature sensors, high-resolution cameras, GPS receivers, and several radios [2].

Figure 1: NASA's PhoneSat Nano-satellite, shown next to a coffee mug for size comparison.

In a typical NASA program, a purpose-built nano-satellite would have cost hundreds of thousands, possibility millions of dollars. Instead, NASA engineers kept the total cost of the components in the nano-satellites to $3,500 by using only the commercial off-the-shelf hardware and keeping the design and mission objectives to a minimum [2].

The key point here is taking a full advantage of consumer-grade off-the-shelf hardware and adjusting the mission objectives. Without trying to benefit advancements in other industries, achieving the low cost hardware target would be extremely difficult both in terms of time and financial resources.

Low Cost Computing Platform

Another interesting development in recent years is the emergence of low cost computing platform, typically in the form of System on Chip (SoC) design where most of the critical components such as CPU, GPU, network connections are incorporated into a very small footprint. Some examples are Aruino, Rasperry Pi, and BeagleBoard products.

Other Influential Concepts

In addition to the main influencers, there are more concepts and advancements worth mentioning.

- Smart Dust:
  Smart Dust typically refers to a system of many very small sensors (usually, microelectriomechanical systems, or MEMS devices) that can detect one or more environmental conditions such as light, temperature or chemicals. They are usually distributed over some area and operated on a computer network wirelessly. An essential concept in Smart Dust is using very small, and very cheap hardware that can be easily replaced.
- Distributed Computing:
  The current trend in development of electronics in the passenger car industry is to reduce the number of Electronic Computing Unit (ECU) hardware. Since a modern passenger automotive may have as many as 80 to 100 ECU's, it is reasonable to attempt to identify potentially redundant hardware and

incorporate as many features in the smallest number of ECU's as possible. This approach is essentially a centralized computing approach. However, the approach taken in this paper follows a different philosophy of distributed computing where local or distributed decision making and the resulting flexibility are preferred.

SYSTEM DEVELOPMENT

Speed limit sign recognition is chosen as the first application area in this project. Hardware, software and development tool environment are chosen based on the system development direction influenced by ideas described above. The goals are re-iterated:

- Hardware that is inexpensive and easily replaceable. This approach increases a system functional up-time.
- Development tool environments that are standard, and/or open source.
- Flexible software environment that allows an easy development process as well as reaching an optimal compromise between performance and resource requirement.

Hardware Setup

A Rapsberry Pi, Model B is used in this project as the computing platform (Figure 2). This platform has the following specifications:

- Broadcom BCM2835 System on Chip
- ARM1176JZF-S 700 MHz processor
- VideoCore IV GPU with 512 MB of RAM
- SD card with 4 GB for booting, OS and data storage

This computing platform is in essence a general purpose computer at a low price point (USD$35 in 2013). For video image capturing, a Logitech Webcam (HD) is used. The overall system setup is shown in Figures 2b and 2c. Note that, in this setup, video images from a single camera may be fed to two Raspberry Pi boards (Figure 2b). An alternative setup with two cameras can also be used at a minimal additional cost (Figure 2c). Two computing boards are planned since alternate frames will be processed by each board. This approach will reduce computing burden on individual computing platform.

Figure 2: Raspberry Pi, Model B.

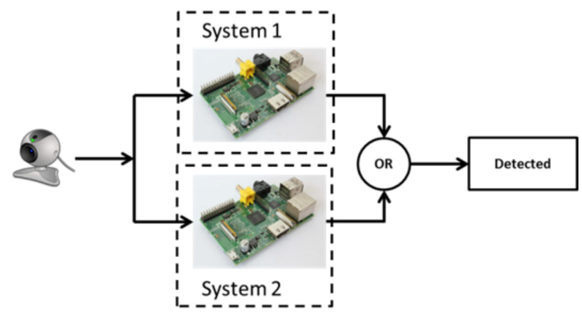

Figure 3b: Hardware Setup with Single Camera.

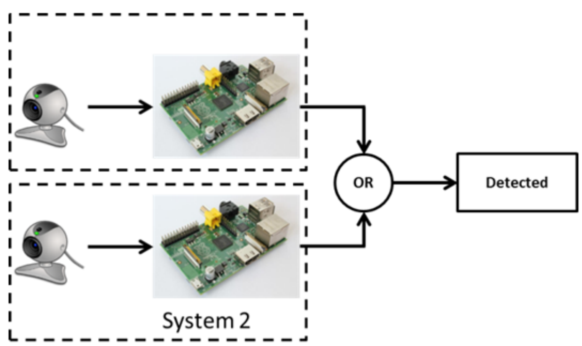

Figure 4c: Hardware Setup with Two Cameras.

Software Setup

Open source image processing and character recognition software packages were used in this project. The algorithm was developed with the one-camera, two-Raspberry PI board setup in mind (Figure 2b). In this simulated setup for algorithm development, video stream from a single camera is processed by two identical processing units.

ALGORITHM DEVELOPMENT

Algorithm for speed sign recognition has been developed on a laptop computer. The algorithm consists of two main parts: speed sign recognition (while rejecting all other road signs) and the actual speed limit recognition. The flow chart of algorithm is shown in Figure 3.

Basic Image Processing

In this step, a frame of image is captured from a stream of video images. Then, a subset of the pixels is divided into areas called Window 1 and Window 2 (see Figure 4a). By focusing on the area of images that is more likely to contain the speed limit sign (the upper, right hand corner in an image), precious clock cycles are not wasted, significantly improving the processing time.

Find Speed Limit Sign Candidate

The usual image processing techniques are used such as converting to grey scale, finding contours and cropping the area of interest in the image.

Optical Character Recognition

Once a potential speed sign is detected through the Support Vector Machine (SVM) algorithm, Optical Character Recognition (OCR) algorithm is run to identify individual number characters.

OR Operation and Decision

Once two sub windows are processed, a simple OR operation is performed: if any one of two image processing systems detects it, it is a proper detection.

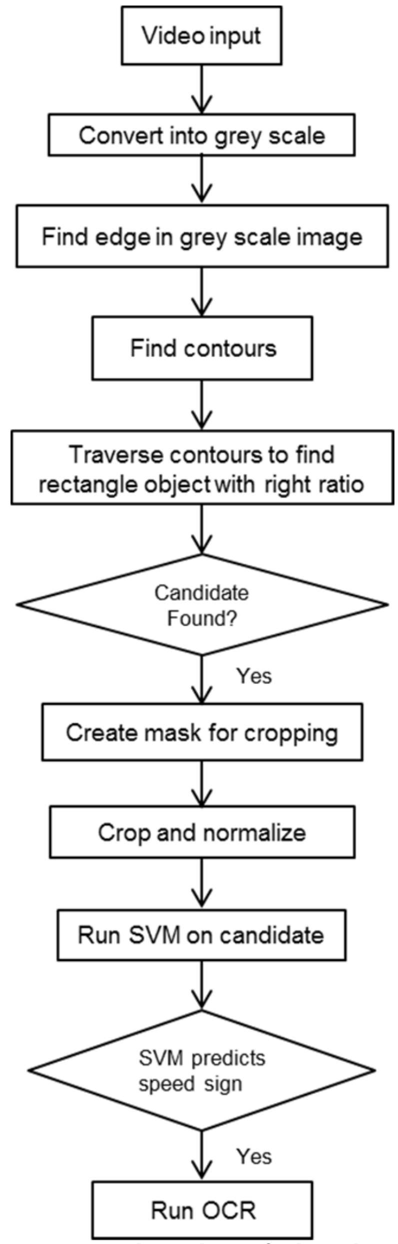

Figure 3. Flow Chart of Algorithm

RESULTS

Three different cases are discussed in this section. The algorithm described above was run in real-time on a laptop computer. The host vehicle was driven at roughly the speed limit.

Case 1

Error! Reference source not found. shows the original still image captured from the video. The host vehicle was traveling at roughly 45 mph, with the speed limit sign showing up to the right of the host vehicle. As discussed previously, the image is divided into three columns, and the right two columns (Figures 3b and 3c, respectively) represent the areas that are separately processed (Figure 4b for system 1, Figure 4c for system 2). In this case, the speed limit sign and the speed limit value (45 mph) were captured and correctly identified in Window 2 while nothing was captured in Window 1.

Figure 4a: CASE 1-Main

Figure 4b: CASE 1-Window 1

Figure 4c: CASE 1-Window 2

Case 2

This case illustrates an opposite situation where a right image was captured by System 1 (Figure 5b) while System 2 missed detection (Figure 5c).

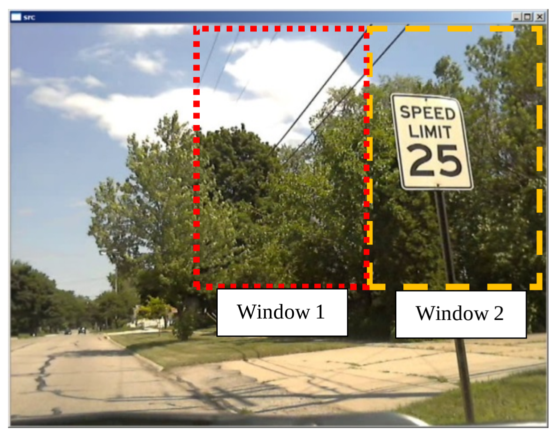

Figure 5a: CASE 2-Main

Figure 5b: CASE 2-Window 1

Figure 5c: CASE 2-Window 2

Figure 6b: CASE 3-Window 1

Figure 6c: CASE 3-Window 2

Case 3

An ideal detection scenario is illustrated in this case. Both the systems 1 and 2 were able to capture the image of the speed limit sign, and properly identity the speed limit of 40 mph. Also note that the no parking sign underneath the speed limit sign was correctly identified as a road sign (indicated by red box around it), but also correctly rejected as a speed limit sign.

Figure 6a: CASE 3-Main

Summary of Results

Overall, experiments under real world and controlled conditions have shown about an 80% detection success rate. In this setup, most of the failures essentially came from missing images from the frame that was captured by the low cost computing platform. While this is not an ideal detection rate, it is more than enough to demonstrate the potential of this unconventional setup.

Limitations

The limitation of the particular low cost general computing hardware has become obvious. On a laptop computer, the algorithm was able to handle the video stream from the web cam at the maximum rate of 30 fps (frames per second). However, the low cost Raspberry Pi board could barely process the incoming raw images at 4 fps. With the image processing algorithms running as discussed above, the boards could handle only about 2 fps even after optimization to speed up processing time. At this frame rate, it is extremely unlikely that any further optimization would improve the performance much.

CONCLUSION AND NEXT STEP

A new approach for ADAS development has been discussed with the goal of keeping the overall development cost as low as possible. The advantage of this approach is numerous: off-the-shelf, low cost computing hardware enables cost-effective deployment of such systems due to off-the-shelf computing platforms that are inexpensive to replace. These commodity computing hardware also benefits tremendously from advancements made by other electronics industries. However, the performance limitations present a challenge, especially for ADAS that are used at high speeds. In light of this constraint, other ADAS features are under development for low speed applications.

REFERENCES

[1] Frost & Sullivan, "ADAS Goes Mainstream: Frost & Sullivan Projects Nearly 40 Per Cent of Compact Vehicles in Europe to Have ADAS by 2018", Frost & Sullivan Press Release, January, 2012.

[2] http://www.nasa.gov/offices/oct/crosscutting_capability/edison/phonesat.html