# Traffic Sign Detection, Recognition and Indication (August 2008)

Greg Browne, *Master of Engineering Student, Memorial University*

***Abstract*—This paper presents a machine vision system that monitors the roadside for speed limit signs using a dash-mounted camera. Images are acquired every two seconds and processed using such image processing techniques as median filtering, automatic thresholding via Otsu's Method, and blob analysis using various moment invariants as form parameters. A display is provided via the model simulation that updates the speed every time a new valid speed is detected. The motivation behind the development is to provide assistance to inexperienced drivers who may benefit from a system that tracks the currently posted speed limit.**

***Index Terms*—driving safety, image processing, machine vision, speed limit sign recognition.**

## I. Introduction

MANY car accidents are due to excessive speeds caused by drivers' inattention to the posted speed limit (Hovde Dassow & Deets LLC., 2007). Moreover, inexperience or improper driving habits may cause the driver to miss a speed limit sign.

The objective of this project is to provide a mechanism that can track the currently posted speed limit and provide a simple display and notification to the driver in order to keep them informed. A minimum number of features for the in-car system are considered a requirement to reduce the amount of attention required of the driver in order to convey the information. Hence, the system should be beneficial to drivers instead of providing a further distraction.

A machine vision approach is the basis of the project, as there is no other currently known method of identifying speed limit signs (such as RFID).

## II. Similar Systems

Similar systems have been developed for traffic sign detection and recognition, but lack the speed limit interpretation aspect.

For example, the system demonstrated by The Mathoworks (2008) is limited by the fact that it can only detect signs that are predominantly red in color. That system gives an indication of *Stop*, *Do Not Enter* and *Yield* signs.

The system developed by S Zhu, L Liu and X Lu (2006) detects traffic signs and labels them as *Regulatory*, *Guide* or *Warning*.

Paper submitted August 11, 2008. This work was supported in part by the Faculty of Engineering, Memorial University.

Greg Browne is a graduate student with the Faculty of Engineering at Memorial University, St. John's, NL A1B 3X5 Canada (phone: 709-743-0306; e-mail: Greg.P.Browne@Gmail.com).

## III. System Platform

The camera used for image capture is a ProLynkz PWC-010 with a maximum resolution of 1.3 MP at 30 fps. Its base is a clip to easily mount it to the pillar or visor of the vehicle.

The camera is connected via USB 2.0 to a laptop with a 2 GHz dual-core processor and 2.5 GB RAM to facilitate the image capture and processing in real-time.

The camera parameters are optimized for outdoor capture with empirically-determined values of 32 exposure value, minimum contrast, 32% gamma correction, and 64 brightness (analogous to the *value* in HSV). These parameters provide a slightly washed out image that correspond to better thresholding results in the algorithm explained in section IV.C.

Furthermore, the camera is configured to use a grayscale colorspace, due to the fact that color is unneeded for the analysis, as well as capture images at a resolution of 640 × 480 to maintain a reasonable object size with a minimum processing time.

## IV. Simulation Model

The machine vision system is modeled in Simulink™, with custom m-file blocks, as shown in Fig. 1. The model is setup on a 2-second timer, such that every two seconds the status is checked and, if it is completed, the simulation is restarted. This timer allowed for a continuous capture of images without interrupting image processing, which is 2-4 seconds on average.

The simulation blocks are described in the following sub-sections.

### *A. Image From Camera*

The *Image From Camera* block is an *Image From Workspace* element that acquires an image from the video object associated with the camera.

### *B. Median Filter*

The *Median Filter* block is a custom m-file element that performs median filtering on the acquired image input from the *Image From Camera* block.

Using the neighborhood shown in Fig. 2, salt and pepper noise is removed without destroying any thin lines or corners present in the image.

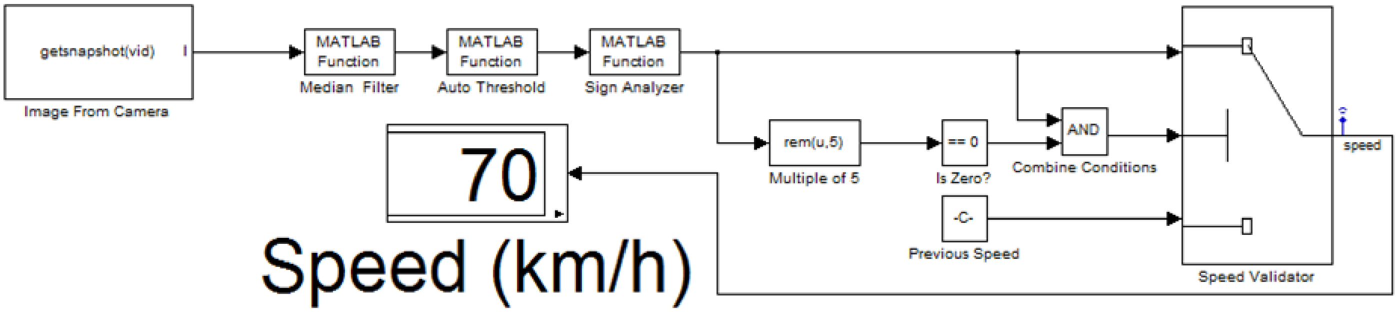


**Fig. 1 Simulation Model**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | **1** | 0 | 0 |
| 0 | 0 | **1** | 0 | 0 |
| **1** | **1** | **1** | **1** | **1** |
| 0 | 0 | **1** | 0 | 0 |
| 0 | 0 | **1** | 0 | 0 |

**Fig. 2 Median Filter Neighborhood**

### C. *Auto Threshold*

The *Auto Threshold* block is a custom m-file element that performs thresholding of the image input from the *Median Filter* block, using Otsu's Method (Otsu, 1979).

Prior to performing Otsu's Method, the algorithm attempts to minimize the threshold interval to improve efficiency. A 4th order polynomial is fit to the image histogram and 1st and 2nd derivative tests are performed on it to determine extrema and associated concavity. If two unique, real peaks are found, the optimal threshold must lie between them – this prevents the algorithm from having to search the entire available gray-level range.

Once the interval has been determined, the optimal threshold is the value that minimizes the in-class variance. Conversely, to simplify computation, the threshold, *T*, is the value that **maximizes between-class variance**, $\sigma^2_{Between}(T)$ (Morse, 2000), as shown in (1).

$$\sigma^2_{Between}(T) = n_B(T) n_O(T) [\mu_B(T) - \mu_O(T)]^2 \quad (1)$$

Where, $n$ is the number of pixels and $\mu$ is the combined mean for the entities subscripted as $B$ for background and $O$ for object.

The resulting threshold is then applied to the image to covert it from grayscale to binary (monochrome).

### D. *Sign Analyzer*

The *Sign Analyzer* block is a custom m-file element that performs contour tracking, blob analysis and speed recognition on the thresholded image input from the *Auto Threshold* block.

Objects in the image are identified using crack code contour tracking and their respective moments calculated ($M_{00}$, $M_{01}$, $M_{10}$, $M_{02}$, $M_{20}$, and $M_{11}$).

The identified objects undergo three stages of processing – pre-conditioning, parametric form analysis and post-conditioning, described in the following sub-sections.

#### 1) *Pre-Conditioning*

Only objects that have an area ($M_{00}$) larger than 16 square pixels are considered as potential digits of the speed. Objects that fail this criterion are considered noise and disregarded.

#### 2) *Parametric Form Analysis*

Objects that pass pre-conditioning undergo parametric form analysis. The form parameters calculated include the *number of holes* (indicated by a negative area when tracking via crack code), *complexity*, *circularity* (Haralick, 1974), $R_I$, $\psi_1$, and $\psi_2$ (Krouglicof, 2007) using (2)-(6).

$$Complexity = \frac{Perimeter^2}{M_{00}} \quad (2)$$

$$Circularity = 1.411 \left[\frac{\mu_R}{\sigma_R}\right]^{0.4724} \quad (3)$$

$$R_I = \frac{\overline{I_{min}}}{\overline{I_{max}}} = \frac{(\overline{M_{20}} + \overline{M_{02}}) - \sqrt{(\overline{M_{20}} - \overline{M_{02}})^2 + 4\overline{M_{11}}^2}}{(\overline{M_{20}} + \overline{M_{02}}) + \sqrt{(\overline{M_{20}} - \overline{M_{02}})^2 + 4\overline{M_{11}}^2}} \quad (4)$$

$$\psi_1 = \frac{\overline{M_{20}} + \overline{M_{02}}}{M_{00}^2} \quad (5)$$

$$\psi_2 = \frac{(\overline{M_{20}} - \overline{M_{02}})^2 + 4\overline{M_{11}}^2}{M_{00}^4} \quad (6)$$

Where, $\mu_R$ is the mean distance from the centroid of the object to its border, $\sigma_R$ is the associated standard deviation, $I$ is the moment of inertia, and an over-bar represents a central value (referenced with respect to the centroid, as opposed to the image origin).

The form parameters, $f_i$, are normalized (0-1) and compared against normalized values, $t_i$, obtained by using training images, such as Fig. 3 (CanLII, 1990). The comparison is based on the least sum of squared deviations, as shown in (7).

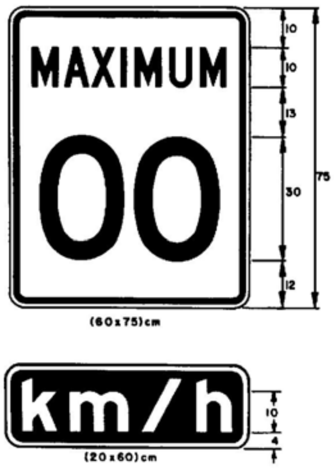


**Fig. 3 Sample Training Speed Limit Sign**

$$Error = \sqrt{\sum_{i=1}^{6}[W_i(f_i - t_i)]^2} \qquad \textbf{(7)}$$

Each form parameter is assigned a weight, $W_i$, that provides the best classification results. The minimum error associated with the calculated form parameters and each digit (0-9) is used to classify the object as the corresponding digit if its certainty is at least 75%.

It is important to note that a zero digit on a given speed limit sign can take one of three different shapes (varying roundness) based on the standard fonts used in Canada. Hence, there are three sets of form parameters to ensure the fidelity of classification.

### *3) Post-Conditioning*

Objects that are classified using their form parameters must also satisfy some practical conditions.

When arranged in descending order by height, consecutive valid digits will have a height within a tolerance of 10%. This criterion prevents extraneous objects from being matched with digits and contaminating the speed interpretation. For example, a light pole could look like a *1*, or a big donut could look like a *0*, however, it is unlikely that another object will be matched of the same height. Moreover, the tolerance is within 10% instead of a hard equality to allow for slight deviation in acquisition, or even in the sign manufacture.

Consecutive valid digits will be vertically aligned to within 50% of their height. This criterion also prevents extraneous objects from being matched with digits and causing speed misinterpretation. For example, when considering Fig. 4 (WoS, 2007), one of the goat antlers will match a 7 via form parameters and relative height to the 2 valid digits, resulting in a speed of 707 km/h. However, because it is not vertically aligned with the other digits, it is dismissed. Furthermore, the tolerance is 50% to allow digits to still be interpreted regardless of reasonable perspective.


**Fig. 4 Speed Misinterpreted Without Post-Conditioning**

There will only be two or three valid digits in an image – realistically, there will not be any speed limits with one digit or more than three digits. This is to prevent an advertisement, or similar sign, from causing the system to interpret a string of numbers (like a phone number) as a speed limit. If less than two digits or more than three digits are recognized, the speed is dismissed.

The interpreted speed must be less than 255 km/h. In practicality, most speed limits will be in the range of 30-110 km/h. If a speed is determined to be higher than 255 km/h, it is dismissed.

## *E. Speed Validator*

The *Speed Validator* block is a *Switch* element that adds an additional post-condition for the run-time speed display.

The *Speed Validator* switches the speed output from the *Speed Analyzer* if it is a non-zero multiple of five. This adds further protection against noise that unstatistically passes all validity tests and presents a false positive. In practice, there are no posted speed limits that are not a multiple of five. Also, the non-zero condition prevents the switch from triggering when no valid speed is found.

If the condition fails, then the last known valid speed is switched back onto the display. This maintains the current speed limit for the driver to review regardless of the distance between speed limit signs.

# V. Typical Sequence

Fig. 5 shows a typical scene from the mounting location of the camera. After undergoing median filtering, auto thresholding and blob analysis, the resulting image and recognition is shown in Fig. 6. The resulting speed is displayed in the *Speed* block of the model simulation, in Fig. 7.


**Fig. 5 Image from Camera**

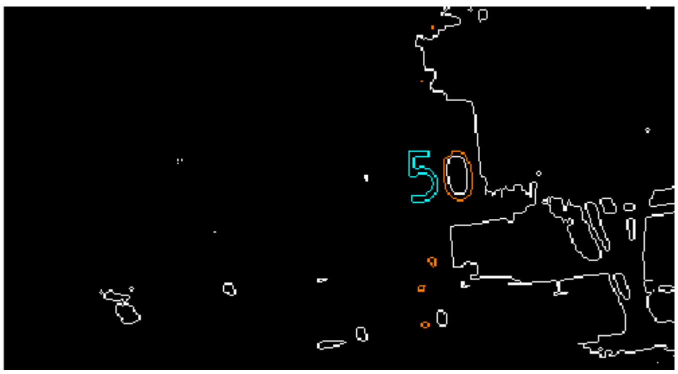


**Fig. 6 Image after Processing & Analysis**

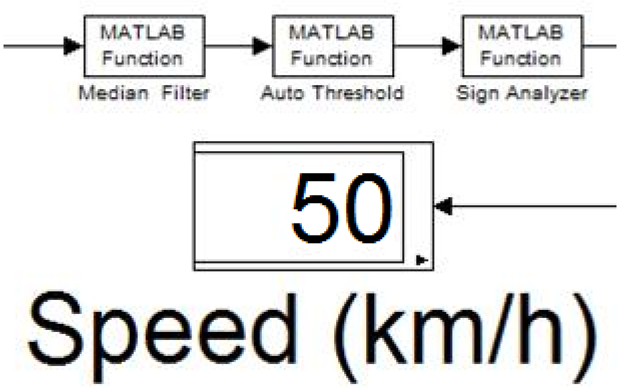


**Fig. 7 Model Simulation Speed Display**

## VI. Future Development

A modular addition to the machine vision system described in this paper would be the integration of an OBD-II interface, such as that previously developed by myself, shown in Fig. 8, for a project comparing fuel economy (Browne, 2008).

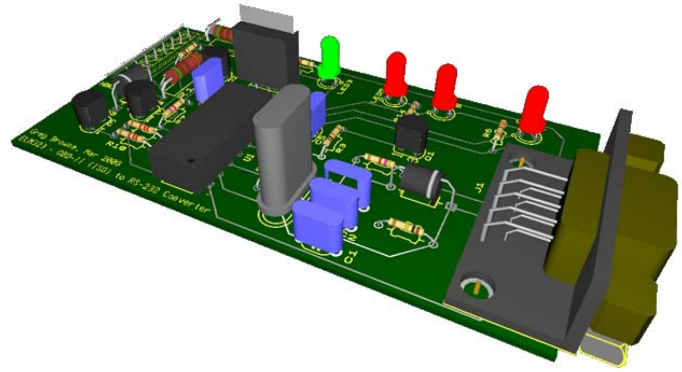

**Fig. 8 OBD-II Interface**

The interface could be added to the simulation model as a custom m-file element that would communicate with the ECU (Electronic Control Unit) of the vehicle every period to request the current vehicle speed. This value could then be compared to the current valid speed limit to provide a notification to the driver if excessive speeding is detected.

Integration of this module would require little effort as previous development has included the implementation of the serial object and communication (including requesting vehicle speed).

## VII. Conclusion

The goal of the machine vision system described in this paper is to provide an aide to inexperienced or careless drivers that could potentially benefit from having the roadside monitored for them.

Young drivers could certainly benefit from a system that would notify them of when they are speeding until they become accustomed to driving.

Driver education has become a significant part of a teenage life in North America and any support technology can provide young drivers could be seen as a potential asset.

With further testing of the algorithm to ensure its robustness, along with the integration of the OBD-II interface to complete the feedback loop, the presented system could perform the function beneficial to these drivers.

## References

Browne, G. (2008). *Design of Experiments to Determine Factors Contributing to Fuel Economy.* Memorial University, Faculty of Engineering, St. John's.

CanLII. N/A. *SIGNS, R.R.O. 1990, Reg. 615.* Canadian Legal Information Institute, Ottawa, Ontario, Canada.

Haralick, R. (1974, January 31). A Measure for Circularity of Digital Figures. *IEEE Transactions on Systems, Man, and Cybernetics* , 394-396.

Hovde Dassow & Deets LLC. (2007, May 29). *Causes of Auto Accidents*. Retrieved May 14, 2008, from Car Accident Lawyers Indianapolis | Indiana Reckless & Inattentive Driving Attorneys Fort Wayne, Gary::

http://www.indycaraccidentlawyer.com/PracticeAreas/Causes-Auto-Accidents.asp

Krouglicof, N. (2007). *Form Parameters: Moments.* Memorial University, Faculty of Engineering. St. John's: Memorial University.

Mathworks. (2008). *Video and Image Processing Blockset - Traffic Warning Sign Recognition Demo*. Retrieved May 15, 2008, from The Mathworks - Accelerating the pace of engineering and science: http://www.mathworks.com/products/viprocessing/demos.html

Morse, B. S. (2000). *Thresholding.* Brigham Young University, Department of Computer Science. Provo: Brigham Young University.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics , 9* (1), 62-66.

WoS. MES1443. *http://www.worldofstock.com.* World of Stock, N/A.

Zhu, S., Liu, L., & Lu, X. (2006). Color-Geometric Model for Traffic Sign Recognition. *Computational Engineering in Systems Applications, IMACS Multiconference , 2*, 2028-2032.

Greg Browne received a B.Eng degree in electrical engineering from the Faculty of Engineering, Memorial University, St. John's, in 2007.
He is currently enrolled part-time in the Master of Engineering program at Memorial University with a research interest in automotive modeling and communication.
He is also currently with Root Four Imagination Inc as a full-time Product Verification Engineer on a project focused on teen driving safety.