ABSTRACT
We present the design of a lane marker detection algorithm suitable for embedded real-time systems. The algorithm was particularly designed for fast execution. The achieved execution time was about 11ms for a search window of 200x90 pixels on a single state-of-the-art Digital Signal Processor (DSP). The algorithm was integrated into an embedded real-time stereo vision sensor. This sensor was part of the sensor suite of the autonomous vehicle RED RASCAL of the team SciAutonics / Auburn Engineering. RED RASCAL participated at the National Qualification Event (NQE) in the DARPA Urban Challenge 2007 in Victorville, CA, USA. The embedded stereo vision sensor did a great job and detected lane markers and lane borders with a high reliability.

KEY WORDS
Computer Vision, Embedded Systems, Robotics, Lane Marker Detection

1. Introduction

Autonomous driving involves a lot of - mostly complex - applications. One of them is Computer Vision, often used for navigation of autonomous vehicles. Algorithms and systems in this field help to identify the drivable road and to avoid obstacles and other critical or dangerous surfaces on the vehicle’s trajectory. Autonomous driving requires real-time systems where the execution times of the algorithms are fast enough for the autonomous vehicle to adjust course and speed accordingly [1].

This paper presents the design of a lane marker detection algorithm suitable for embedded real-time systems. The algorithm is an extension of the algorithm presented in [2] and [3], with some redesign and several improvements in design and implementation to cope with the required performance improvements.

The improved algorithm was embedded in an embedded real-time stereo vision sensor. It was part of the sensor suite of RED RASCAL and contributed to the autonomous driving capabilities of this vehicle. RED RASCAL was a vehicle which had been developed for the participation in the DARPA Urban Challenge 2007 and is described in section 3. More details about the competition itself are outlined in section 2. These sections are followed by section 4 where our embedded stereo vision system is described in detail. The description includes implementation and optimization of the lane marker detection algorithm, too. Section 5 summarizes the achieved results. The following section 6 concludes the paper and presents a short outlook on future work to be done.

2. DARPA Urban Challenge

The DARPA Urban Challenge 2007 was the third in a series of competitions of the Defense Advanced Research Projects Agency (DARPA) in the USA to foster the development of autonomous robotic ground vehicle technology for various - mostly military - applications [4].

The focus of the 2007 event was shifted from desert to an urban environment. The urban area selected was the (abandoned) housing area at the former George Air Force Base in Victorville, CA, USA. The challenge for the participating teams was to complete a complex 60 miles urban course with live traffic which had to be completed in less than six hours. During the course, the vehicles had to cope with traffic intersections, live traffic, road following, 4-way-stops, lane changes, three-point turns, U-turns, blocked roads, and pulling in and out of a parking lot [5]. The vehicles reached a speed of up to 30 mph during the race. This was an indicator why real-time was an essential factor for the systems used. Vehicle restrictions were given by the DARPA officials. The vehicle had to weigh between 2000 lbs and 30000 lbs and had to have a wheelbase of 6 ft or more. The height was limited with 12 ft and the width with 9 ft.

Figure 1. RED RASCAL (left) and the Team SciAutonics / Auburn Engineering (right)
Of the 89 teams which initially applied, 35 teams were invited to enter the National Qualification Event (NQE) and to compete in a rigorous eight-day vehicle testing period. The teams had to demonstrate the safety features and robust driving of the vehicles in different situations and scenarios. Out of these 35 teams, only 11 teams managed to qualify for the final event [6, 7, 8].

3. The Vehicle RED RASCAL

The team SciAutonics / Auburn Engineering successfully participated in the Grand Challenges 2004 and 2005 with their ATV-based vehicle RASCAL [9].

For 2007, the team built a new vehicle based on the lessons learned from the former challenges. The new vehicle was a modified Ford Explorer and was adapted for "drive-by-wire". The vehicle remained "street legal" and readily drivable on public highways. It was called RED RASCAL to reflect its close relation to its predecessor. The vehicle and the team SciAutonics / Auburn Engineering are shown in Fig. 1.

3.1 Vehicle Architecture

The high-level architecture of RED RASCAL is shown in Fig. 2. The vehicle is at the bottom. On the lower right, sensors send data up to a series of processing stages. On the lower left, actuators on the vehicle receive commands from the control logic. In the center is the Vehicle State Estimator, which continually makes optimal estimates of the vehicle’s position, velocity, heading, roll, pitch, and yaw, based on inputs from the GPS (Global Positioning System), IMU (Inertial Measurement Unit), vehicle sensors, and obstacle sensors. The vehicle state estimates are used by all parts of the control system, including the Path Planner.

At the top of the diagram is the Mission State Estimator. It determines what phase of a mission the vehicle is in and identifies when unique behaviors such as passing or parking are required. When the vehicle is driving in an urban environment, the key challenge to be addressed is situational awareness. Not only must the vehicle follow a path and avoid moving obstacles, it also must demonstrate correct behavior depending on the situation. The estimator treats each type of situation as a state and uses state diagrams to cope with behaviors and transitions [10].

3.2 Sensor Suite

Sensors were needed for both localization (GPS, IMU, etc.) and perception (obstacle and road detection). A strategy of redundancy was employed to provide measurements from some sensors when others were not available or in the case of the failure of a particular sensor. The main sensor for vehicle localization was a single antenna Navcom SF-2050 DGPS (Differential GPS) receiver with Starfire satellite-based corrections provided by Navcom. A Honeywell HG1700 tactical grade six degree of freedom IMU was used to measure translational accelerations and angular rates. A NovAtel Beeline RT20 dual antenna GPS system was used for obtaining the initial vehicle orientation, as well as longitudinal and lateral velocities [10].

The vehicle used two types of environmental sensors for obstacle and vehicle avoidance: LIDAR (Light Detection and Ranging) sensors and the embedded stereo vision sensor. The capabilities of these sensors overlapped to provide the redundancy desired to protect against sensor failure and improve reliability of measurements. The embedded stereo vision system was used as near- and mid-range (13 ft to 65 ft) obstacle sensing system.

4. Embedded Stereo Vision Sensor

Beside detecting obstacles up to a distance of 65 ft, the embedded stereo vision sensor also provided information about lane markers and lane borders. However, the needed calculations for the algorithm were very time consuming and it was very difficult to fulfill the real-time requirements. The original algorithm needed about 45ms for a calculation of a frame – leaving only 45ms for the obstacle detection algorithm. Therefore, we decided to redesign the algorithm to significantly improve the performance and to use the freed processor budget for obstacle detection to increase its robustness and reliability.

4.1 System Concept of the Sensor

The stereo vision sensor consisted of a pair of Basler A601f monochrome cameras with a resolution of 656 (H) x 491 (V) and a quantization of 8 bits/pixel. Both cameras were connected by a 400MBit-FireWire interface to the embedded vision system. The frame rate of the sensor was 10 fps (frames per second) to cope with the real-time requirements of the vehicle. Of the available calculation time of 100ms, about 90ms were allocated to the vision algorithms...
and the remaining 10ms were used by system tasks of the embedded vision system.

The embedded system was based on a Texas Instruments TMS320C6414 Digital Signal Processor (DSP) running at 1GHz. The operating system was DSP/BIOS II from Texas Instruments. The embedded vision system was responsible for the synchronous acquisition of both images, for the execution of the computer vision algorithms, and the communication with the vehicle central brain via an Ethernet interface using UDP (User Datagram Protocol) sockets [11]. The stereo head of the sensor was mounted on the dashboard while the DSP box was located in the trunk of the autonomous vehicle.

The stereo vision sensor included a debugging interface for real-time logging of the left sensor input image. Extracted obstacles, lane markers, lane borders, and some internal states were logged for field-testing and evaluation, too.

### 4.2 Obstacle Detection Algorithm

The main task of the embedded stereo vision sensor was the detection of obstacles, lane markers, and lane borders in front of the vehicle. For obstacle detection, a fast stereo matching method was used. Furthermore, the bouncing of the vehicle was predicted and compensated in the stereo images to improve the detection and classification of obstacles. A detailed description of this obstacle detection algorithm was presented in [9] and [11].

### 4.3 Lane Marker Detection Algorithm

This lane marker detection algorithm was an extension of the algorithm presented in [2] and [3]. For extraction of lane markers and lane borders, only the left camera image was used. The original version of this algorithm was able to detect lane boundaries, but with low speed and had problems with weak edges, too. To overcome the performance bottlenecks and quality problems and to push the algorithm to achieve real-time performance, we redesigned the algorithm. The layout of the new and improved version of this algorithm is shown in Fig. 4.

For the extraction of lanes, five independent steps were necessary. First of all, all potential edges were enhanced using a Finite-Impulse-Response (FIR) filter [12]. In the second step, the image was converted into a binary image. The threshold for this operation was calculated automatically for each image using the Otsu algorithm [13]. In the next step, the binary image was searched for lines – which represented possible lane markers – using the Hough transformation [14]. Local maxima in the Hough matrices were an indicator for lines with specific parameters. Here, it was important to identify candidates for possible lanes in the image. In the last step, all found lines were analyzed by a decision engine to eliminate all those which were not part of the lane boundaries due to reasoning.

The difference equation of a FIR filter of n-th order is

$$y[k] = \sum_{i=0}^{n} \beta_i \cdot u[k - i],$$

(1)

where $$y[k]$$ is the output signal, $$u[k - i]$$ is the input signal on position $$k - i$$, and $$\beta_i$$ denotes the n filter coefficients. Since we were looking mainly for vertical lines (since we were interested in straight lanes in front of the vehicle – and not in lanes crossing our way), Eqn. 1 can be rewritten as

$$P_{out} = \sum_{i=0}^{n} \beta_i \cdot P_{in}[x - i, y],$$

(2)

where $$x$$ is the column index and $$y$$ is the row index. To achieve a very fast implementation we decided to use a 3\textsuperscript{rd} order filter with coefficients $$\beta_0 = -1$$, $$\beta_1 = 0$$, and $$\beta_2 = 1$$. Therefore, the calculation was reduced to a single substraction.

$$P_{out}[x, y] = P_{in}[x, y] - P_{in}[x + 2, y]$$

(3)

Fig. 5 presents sample results of a street scene of an FIR-filtered input image.

The goal of the auto threshold block was to convert the gray-scale image into a binary image. All pixels with gray values below a threshold were set to zero, all others were set to one. The most important operation was to find a new optimal threshold for each new image. Based on this threshold, the image could be converted into a binary one. We used a technique introduced in [13] to calculate an optimal threshold. By using this threshold, the image was divided into two classes and the within-class variance $$\sigma_w^2(t)$$ was minimized.

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

(4)

Weights $$w_i$$ are probabilities of the two classes separated by a threshold $$t$$ and $$\sigma_i^2$$ are the variances of these classes. This optimal threshold $$t$$ was used to divide the image into a class containing potential lane border elements and a background class.

$$T(P(x, y)) = \begin{cases} 
1 & \text{If } P(x, y) \geq t \\
0 & \text{If } P(x, y) < t
\end{cases}$$

(5)

The thresholding was implemented using the packed data processing feature of the DSP [15]. Using that processing method four pixels were processed in parallel.
The Hough transform is a feature extraction technique, classically used for the identification of lines, but has been extended to identify arbitrary shapes, too. For lines, each line can be represented with the parameters \( r \) and \( \Theta \) \[14\]. Each point in the original image represents a sinusoidal curve in the 2-dimensional Hough space \((r, \Theta)\). Therefore, all found edge pixels (edge elements) result in additional sinusoidal curves in the Hough space. In the Hough space intersections are accumulated and local maxima are searched for. A local maxima represents a potential line in the original image. In Fig. 7 an example of a Hough space, some found maxima, and the corresponding lines in the original image are shown. A more detailed description of the Hough transform and the local maxima search can be found in \[2\] and \[3\].

Finally, the Decision Engine analyzed and eliminated all those lines, which were not part of a lane. Assuming that the vehicle was on the lane, the line parameters could be limited as shown in Fig. 8. Based on a lot of field experiments with the vehicle we experimentally determined best values for \( \alpha \) and \( d \),
\[
\begin{align*}
\alpha & \leq 15^\circ \quad (6) \\
\frac{d}{w} & \geq \frac{w}{4} \quad (7)
\end{align*}
\]
where \( w = 7 \text{ft} \) is the width of our vehicle RED RASCAL. We used a method introduced in \[16\] to transform the planar road scene into the image. This model of the lane border and the given parameters had been proven useful for our application.

### 5. Results

We integrated the new modules into our embedded stereo vision sensor. Furthermore, we did a code proofing to compare the achieved performance with the original one presented in \[2, 3\]. The results of the proofing are shown in Tab. 1. As shown here, the algorithm is now approximately four times faster than before, thanks to the new method of edge extraction and thresholding. For future optimizations, both the Hough transform and the find local maxima method are now candidates for further performance optimizations, since now they consume more than 90% of the overall processor budget.
During the qualification runs for the Urban Challenge event at Victorville, CA, USA, the embedded stereo vision sensor was used for obstacle and lane marker detection. Fig. 9 presents an exemplary output of detected lane markers and obstacles on a road with live traffic.

Due to an accident in the second test run leading to some serious mechanical damages on the vehicle, RED RASCAL was not able to qualify for the final event on November 3, 2008, the Urban Grand Challenge. However, our embedded stereo vision sensor did an excellent job during all pre-tests and test runs. Our sensor had proven to be a valuable assistance system for autonomous vehicles.

6. Conclusion

This paper presented a lane marker detection algorithm used in an embedded stereo vision sensor. The algorithm was designed to detect lane markers and lane borders in front of an autonomous vehicle. Furthermore, it had to fulfill real-time requirements, too. The algorithm was integrated – as part of the sensor – into the autonomous vehicle RED RASCAL from the team SciAutonics / Auburn Engineering. The vehicle participated at the National Qualification Event (NQE) at the DARPA Urban Challenge 2007 in Victorville, CA, USA. The embedded stereo vision sensor did a great job and detected lane markers and lane borders with a high reliability.

Since most of the processing power is now occupied by Hough transform and finding local maxima, a deeper analysis of both blocks would make sense to reduce the processing time. The Hough transform itself could be replaced by another one, like the Randon transform, to improve both the performance and the quality of the line detection (e.g., to better cope with noise artifacts).

Additionally, we have just recently started two new projects for advanced driver assistance systems in next generation intelligent cars. In these projects we plan to disseminate and extend the used technology for next generation intelligent sensor devices.
<table>
<thead>
<tr>
<th>Module</th>
<th>New approach</th>
<th>Old approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR filter</td>
<td>0.489ms</td>
<td>34.5ms</td>
</tr>
<tr>
<td>Otsu method</td>
<td>0.226ms</td>
<td></td>
</tr>
<tr>
<td>Thresholding</td>
<td>0.023ms</td>
<td></td>
</tr>
<tr>
<td>Hough transform</td>
<td>6.938ms</td>
<td>6.9ms</td>
</tr>
<tr>
<td>Find local maxima</td>
<td>3.400ms</td>
<td>3.4ms</td>
</tr>
<tr>
<td>Total</td>
<td>11.076ms</td>
<td>44.8ms</td>
</tr>
</tbody>
</table>

Table 1. Comparison of the performance of the modules of the lane marker detection algorithm. The old approach used a different way of edge detection. Thus, only the sum of the calculations times for that part can be given. The decision engine was not used in the original algorithm. Therefore, no comparison is performed here.

Acknowledgments

The authors would like to thank Mr. Daniel Baumgartner and Mr. Christoph Schönegger from the University of Applied Sciences Technikum Wien, Austria and all people of the SciAutonics / Auburn Engineering team, Thousand Oaks, CA, USA for their support during this project.

The research leading to these results has received funding from the European Community’s Sixth and Seventh Framework Programs (FP6/2003-2006, FP7/2007-2013) under grant agreements n° FP6-2006-IST-6-045350 (robots@home) and n° ICT-216049 (ADOSE).

References


