GPU and CPU Cooperative Accelerated Road Detection

Peng Xiong[1], Cheng Xu[1], Zheng Tian[1] and Tao Li[1]
[1]School of Computer and Communication, Hunan University
Changsha 410082, Hunan Province, China

Abstract - *In this paper, we propose a fast and robust unstructured road detection method that integrates GPU (Graphics Processing Unit) and CPU implementations. In order to ensure the robustness of the algorithm, BP (Back Propagation) Neural Network is employed to learn the color features from a set of sample of both road region and off-road region, and then to classify a newly pixel. And the B-spline curve model is employed to fit the boundaries of the lanes with the Least Square Method. To improve the real-time capability, the NVIDIA CUDA (Compute Unified Device Architecture) framework is used, and a GPU and CPU cooperative acceleration technique is proposed. Taking the advantages of these properties, the proposed implementation works out with high performance of detection in various environments. Meanwhile it is robust against noise, shadows and illumination variations. Moreover, it can performs about 10 times faster than a conventional implementation running on a CPU.*

Keywords: Neural Network, Road Detection, CUDA

1 Introduction

Road detection is one of the most important technologies in the vision-based intelligent navigation system, and is of high relevance for autonomous driving, road departure warning, and supporting driver-assistance systems such as vehicle and pedestrian detection [1]. A robust road detection algorithm should provide accurate road position and direction information for the navigation system [4]. However, conventional detection methods are of high computational costing, and cannot adapt to various environments. Therefore, it is a critical issue to search for a real-time and robust road detection approach, to improve the vision-based intelligent navigation system for a practical application.

Conventional unstructured road detection algorithms could be divided into two groups, the model-based and the feature-based.

The model-based method begins with the hypothesis of the road model, and then matches the road edge with the road model. A B-Spline based lane detection and tracking algorithm, proposed by Yue Wang et al. in [2], is a representative of this kind of methods. It can work without any cameras' parameters. Moreover, the algorithm is robust against noise, shadows, and illumination variations. Still, the result of such method is dependent on the hypothesis of the road model, so they cannot fit to the situation that the shape of road changes greatly.

The feature-based method divides the image into the road region and the off-road region based on the differences of gradient, color or texture between them. Compared with the model-based method, the feature-based method is not sensitive to the shape of the road. Hui Kong et al. have presented a method based on the texture feature of a single image in [3]. Another typical example of this kind of methods is an adaptive road detection algorithm based on BP neural network presented by Mike Foedisch et al. in [5]. The approach extracts the features of a number of road images and trains the neural network based on the feature vectors, and then computes the result of classification. It avoids manual annotation of images by taking advantage of the conforming structure in the road images, thus can adapt to various environments. But the algorithm still has the defect that it is not robust enough against to shadows. Moreover, in order to achieve real-time processing, it processes every other pixel in the image, which has reduced the processing accuracy. In conclusion, it is a huge challenge to meet the real-time constraints while ensuring the robustness for the high computational costing of feature-based methods.

In order to solve the disadvantages mentioned above, in this paper, we propose an approach based on BP Neural Network and B-spline curve, which takes the advantages of the model-based and the feature-based methods. Meanwhile we specifically construct a GPU and CPU cooperative acceleration technique to support real-time and high-performance detection.

In the rest of the article, we firstly introduce the B-spline curve and BP neural network in Section 2. Secondly, in Section 3, we present the detail of our algorithm. And then we present our GPU and CPU cooperative implementation in Section 4. Finally, we describe the experimental results in Section 5.

2 B-spline Curve and BP Neural Network

B-spline curve is a commonly used model in road detection, it is a linear combination of basis curves. Unlike other road model, a B-spline model has more advantages as local control that the degree of a B-spline curve is separated from the number of control points, and control flexibility that change the position of a control point without globally changing the shape of the whole curve [6].

However, as other model-based road detection methods, the result of B-spline curve model-based algorithm is dependent on the hypothesis of the road model. That is to say, this method is not robust enough against changing environment, such as when the road surface is not level, the method will make a big deviation.

Color appearance information has been widely used as the main cue for road detection, since color provides powerful information of the road to be detected in the absence of reliable shape information. In addition, color imposes less physical restrictions, leading to more versatile systems [7]. Therefore, we use neural network to segment road and off-road regions after learning the color features of the image. As a result, it overcomes B-spline curve's frangibility against the interference of the changing shapes of the road.

Neural network is a way to learn the nonlinearity at the same time as the linear discriminant. Such multilayer networks can provide the optimal solution to an arbitrary classification problem. The key power provided by such networks is that they admit fairly simple algorithms where the form of the nonlinearity can be learned from training data [8].

As a result, combing neural network with B-spline curve road model can take both advantages of these two methods. It can accurately divide the image into road region and off-road region, and then quickly fit the road boundary through the result of classification. This method is robust against shadows, illumination variations, and the changing road shapes.

3 The Road Detection Approach

Our algorithm mainly consists of five phases.

3.1 Classification

In the step, we classify every pixel using a BP Neural Network, which has been trained by samples of road region's and off-road region's color features.

The neural network presented consists of three layers, an input layer, a hidden layer, and an output layer. They are interconnected by links, which contain modifiable weights, between layers.

We convert the image to HSV (Hue, Saturation, Value) color space, and use H and S value as an input vector of the network. The input vector is presented to the input layer, each hidden unit performs the weighted sum of its inputs to form its net activation, the net activation can be written as:

$$net_j = \sum_{i=1}^{d} x_i \omega_{ji} + \omega_{j0} = \sum_{i=0}^{d} x_i \omega_{ji} = \boldsymbol{\omega}_j^t \mathbf{x} \qquad (1)$$

where the subscript i indexes units on the input layer, j for the hidden, ω_{ji} denotes the input-to-hidden layer weights at the hidden unit j. Each hidden unit emits an output that is a non-linear function of its activation: $y_j = f(net_j)$.

The net activation and emits of the output layer units net_k and z_k is computed similarly based on the hidden unit signals. Then we use z_k to classify the input pixel as road or off-road.

3.2 Block segment

Generally, the pixel in road and off-road region are continuous and similar, so we can divide the image into blocks, and then classify blocks by its four corner regions pixels. In actuality, the influence of noises can be reduced by the approach. The method is described as follows.

Suppose a pixel x belongs to either road area R or off-road area NR, that is $x \in \{R, NR\}$. And the corner region C belongs to road area or off-road area, which is represented as: $C \in \{R, NR\}$ .The probabilities of a corner region belonging to the road area or off-road area can be computed as:

$$p(C \in R) = \sum if(x_i \in R)/N \qquad (2)$$

$$p(C \in NR) = \sum if(x_i \in NR)/N \qquad (3)$$

where N is the number of pixel in a corner.

If four corner regions in the block all belong to road area or off-road area, mark this block as road block or off-road block. The membership probability is defined as:

$$p(block \in R) = \sum p(C_i \in R)/4 \qquad (4)$$

$$p(block \in NR) = \sum p(C_i \in NR)/4 \qquad (5)$$

If some corner regions in a block belong to road area, while others belong to off-road area, mark the block as mixed block, and the membership probability $p(block \in MIX)$ is defined as the mean of the probabilities of four corners belong to the corresponding areas.

Now, the image can be regarded as a matrix of blocks, with each block marked as Road, Off-road or Mix. In some case, the blocks might be misjudged by the influence of noise. To deal with such error, we scan the blocks row by row. When we meet a Mix block, and the left block and the right block are marked as the same, then we test the probability of these three blocks. A block with a probability lower than a threshold will be reconfigured.

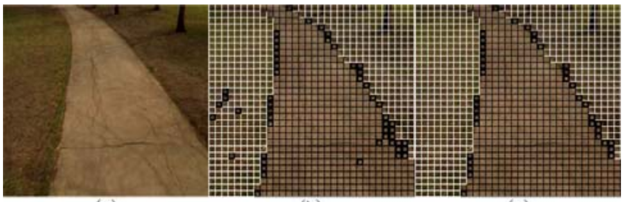

Fig. 1 Result of Classification

Fig. 1 shows the results of classification. (a) is the original image. (b) is the primary classifying result in which the thick-black blocks are mixed block. (c) shows the final result of classification which is modified by error fix based on the membership probability.

3.3 Edge Extraction

It can be sure that the current road boundaries exist only in the Mix blocks. Then we extract the road boundaries form these blocks using an approach as follows:

- *Scan the blocks from the bottom to the top. For each row, classify the midline pixels of Mix blocks from left to right. Take the continuous road pixels set in the scan line to be a candidate sub-segment line.*

- *Merge the close candidate road sub-segment line. Deal with all scan line's candidate road's sub-segment, and get the road line set.*

- *Finally, for the road segments on each scan line, extract the line segment's left and right boundary points, and then obtain the boundary points set of the road.*

3.4 Fitting

Due to its advantage of making the construction of curves with high stability, the B-spline curve is chosen as our road model. A cubic B-spline curve with $n+1$ control points $P_i (i = 0,1,\cdots,n)$ can be expressed as:

$$C(u) = \sum_{i=0}^{n} P_i N_{i,4}(u) \tag{6}$$

where $N_{i,4}(u)$ is the base function [9], and the matrix format is:

$$C(u) = \left[u^3, u^2, u, 1\right] \times \frac{1}{6} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 0 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} P_{i-1} \\ P \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \tag{7}$$

In practice, most roads have only one or two turning corners in an image, so, three control points to obtain B-spline curve model is robust enough for the navigation system. In our approach, we select the first and the last control points of the interpolation sequence $\left[P_1, P_2, \cdots, P_n\right]$ as the first interpolation point and the end interpolation point. Then, we use the Least Square Method to search the interpolation sequence $\left[P_2, P_3, \cdots, P_{n-1}\right]$ and select the optimized position of the second control point [4]. Making use of the accurate result of edge extraction, small numbers of points need to be searched for. As a result, the fitting process is very fast and robust.

3.5 Updating

To adapt to the changing environment, it is necessary to update the network weights. After the B-spline fitting phase, the image had been divided into road region and off-road region. Then we choose one block in road region and one block in the off-road region as samples for updating. In order to keep the network from the influence of noise, we agree on that the membership probability $p(block \in R)$ or $p(block \in NR)$ of the block chosen as sample is higher than a threshold.

Similar to the training phase of the network, the update rule for the hidden-to-output weights can be presented as follows:

$$\Delta\omega_{kj} = \eta\delta_k \frac{\partial net_k}{\partial\omega_{kj}} = \eta(t_k - z_k) f'(net_k) y_j \tag{8}$$

where η is the learning rate, δ_k is the sensitivity of unit k: $\delta_k = -\partial J / \partial net_k$.

The learning rule for the input-to-hidden weights is:

$$\Delta\omega_{ji} = \eta x_i \delta_j = \eta \left[\sum_{k=1}^{c} \omega_{kj}\delta_k\right] f'(net_k) x_i \tag{9}$$

where δ_j is defined as the sensitivity of unit j : $\delta_j \equiv f'(net_j) \sum_{k=1}^{c} \omega_{kj}\delta_k$.

4 GPU & CPU Cooperative Processing

In the classification phase, for every pixel, the net activation and emission of each unit in each layer need to be computed. It is clear that it needs much more computational cost with the heavy matrix computation. What's more, it is impossible for a general CPU implementation to handle in real time as the image size grows.

GPGPU (General-Purpose computation on GPUs) enables real-time processing for an algorithm requiring huge computational cost [10]. Thus, we present a CPU and GPU cooperative acceleration technique to support real-time road detection. In this section, we describe the detail of our GPU and CPU integrated implementation.

Our approach can be split into two parts: the host and the GPU processing.

On the CPU (host), the image is acquired and copied into GPU texture memory for fast access from the GPU. The weights of the neural network, which had been trained previously, are copied into GPU constant memory, in which the data could be accessed in only one GPU cycle in the ideal case. Since the weights of the neural network stay unchanged during the classification phase, such implementation could largely speed up the classification process though the constant memory cache.

On the GPU, the image pixels are classified into two classes, and the result of classification phase is sent to CPU.

Then in the edge extraction, the B-spline fitting and the network updating phase are finished on CPU. The block diagram can be expressed as Fig. 2.

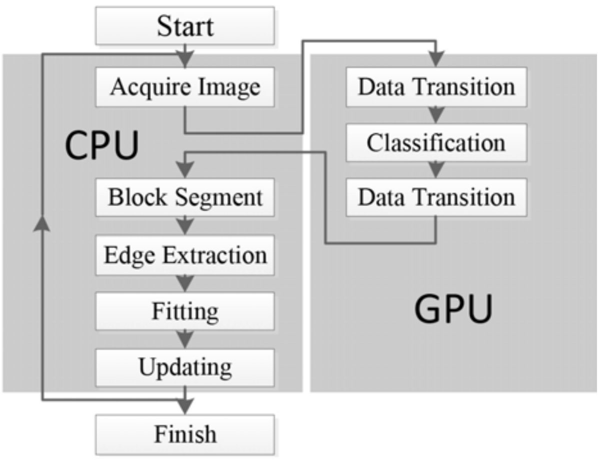

Fig. 2 The Algorithm Block Diagram

With the GPU-accelerated processing, the classification phase, which takes much computational cost, is accomplished by the GPU. Such a technique can speed up the processing course, but could be further promoted.

Since the edge extraction phase needs the result of classification which is produced by GPU, CPU has been waited in vain for the outcomes of GPU. On the other hand, while CPU starts with the edge extraction and fitting phase, GPU is in idle state. Due to the waiting process between CPU and GPU, both of them are not been fully used. Moreover, the transmission delay between CPU and GPU makes the situation worse for it deducing the benefits of GPU acceleration greatly. It can be described in Gantt chart in Fig. 3.

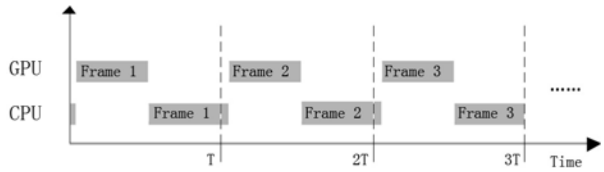

Fig. 3 Implementation with GPU-accelerated Processing

To deal with such problem, we present a GPU and CPU cooperative accelerated implementation, which is based on the basic idea of the pipeline. Such implementation can eliminate the waiting process, and further improve the processing speed. The detail of the technique is as follows:

An image is acquired on CPU and copied into GPU memory, and then GPU starts to process the image. While the classification process is accomplished, the result is sent to CPU. Then the CPU prepares the next frame, which is to be processed, and copy it into GPU memory. Then the edge extraction and following phase are processed on CPU, and the classification phase of next frame is processed on GPU in parallel. Such implementation could be described in Gantt chart in Fig. 4.

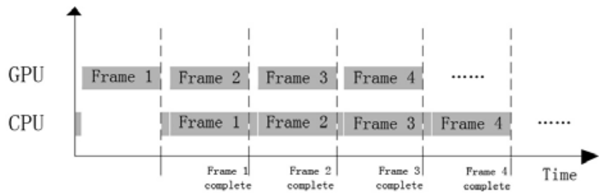

Fig. 4 GPU & CPU Cooperative Accelerated Processing

In practical application, different images generally take identical time cost on GPU, but the time consumption on CPU might vary depend on the complexity of the result of classification. The cooperative accelerated processing technique can maximize the use of both CPU and GPU, and efficiently reduce the waiting process. Moreover, such technique can conceal the delay of data transition between CPU and GPU and ultimately boost the peak performance. By using this technique, the propose implementation can performance in real time. The performance is presented in section 5.2.

5 Experiments

In this section, we will demonstrate the result of our algorithm performed on a personal computer equipped with NVIDIA GPUs. The proposed approach was realized using the lib of Open source Computer Vision library. We will show the efficiency and real-time capability of the proposed algorithm.

5.1 Effectiveness

In order to show the effectiveness of the proposed algorithm, we design two groups of experiments, one group is in the case of shadows and illumination variations while the other is with various environments. Due to the limitations of the experiment environment, we carry out a variety of simulation experiments on video images provided by the Vision and Automation System Center in Carnegie Mellon University instead of field tests. These sets of road pictures can be downloaded from http://vasc.ri.cmu.edu/idb/images/.

Fig. 5 demonstrates a part of results in our experiments. The first row shows the results of Group One where the experiments are carried out in the environment of shadows and illumination variations, and the second row is the results of Group Two where the experiments are carried out in different environments, such as snowy, rainy and fallen leaves. In the Fig. 5, (a) is a scene with shadows, the trees on the two sides project shadows on the road surface. (b) is a noon scene where there is great strong sunlight. (c) is a dusk scene where it is some kind of dark. (d) is a snow scene, the road surface is covered with snow. (e) is a scene after the rain, the boundaries between road and off-road regions are very fuzzy. (f) is a fall scene, both sides of the road is covered with fallen leaves. Clearly, our algorithm can accurately extract the boundary of the road. In scene (a), the road is covered by shadow, the block-classifying method used in our algorithm can exclude the interference of noise and accurately accomplish the fitting. In scene (e), it is difficult to distinguish between road and off-road regions even by eyes. However, our algorithm can still do perfect classification and accurately find the boundary of the road.

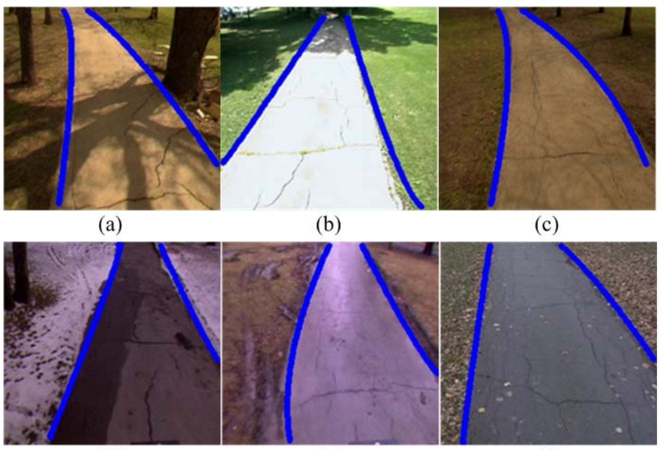

Fig. 5 Experiments

In order to evaluate the performance of our approach, we employ a pixel–wise measures from which three error measures are computed: quality, detection rate and detection accuracy [7], see Table 1 and Table 2. Five group of images advantage different environment are manually segmented to generate the ground-truth.

Table 1 Pixel–wise Measures Definition

Contingency Table		Ground-Truth	
		Off-Road	Road
Result	Off-Road	TN	FN
	Road	FP	TP

Table 2 Evaluation for the Performance of Detection Results

Pixel-wise measure	Definition
Q(Quality)	$\frac{TP}{TP+FP+FN}$
DR(Detection Rate)	$\frac{TP}{TP+FP}$
DA(Detection Accuracy)	$\frac{TP}{TP+FN}$

The performance of the proposed method is validated and compared to the algorithm based on kernel density estimation introduced in [4].

Our method can keep high and stable performance in various environments. Especially, our method is little interfered by shadows and illumination variations and the detection accuracy even reaches 98.4%, which we can clearly see from Table 3.

Table 3 Performance of Our Algorithm

	Kernel Density Estimation			Our method		
Shadows	Q	DR	DA	Q	DR	DA
Fall	0.747	0.909	0.784	0.967	0.982	0.984
After Rain	0.751	0.891	0.791	0.909	0.933	0.966
Snow	0.749	0.894	0.797	0.907	0.969	0.937
Night	0.689	0.824	0.782	0.774	0.903	0.81
Night	0.729	0.906	0.775	0.751	0.854	0.827

Only in the case of night that there is little difference between the hue and saturation of road and off-road regions, detection rate of our method might descend.

5.2 Real-time Capability

In order to demonstrate the effect of our acceleration technique, we design two group experiments.

We choose a video sequence which contains 35 images to run on two desktop for 30 times in each experiment. The average time consumption of each image is listed in Table 4.

Compare with Serial algorithm, the GPU-accelerated algorithm can reduce time consumption greatly. And through reducing the waiting process and transmission delay, our CPU and GPU cooperative acceleration technique could boost the performance further.

Table 4 Time Consumption Comparison

$\left({ms}/{frame} \right)$		Block size 8	Block size 16
Image size 256*240	Serial	74.2	24.2
	GPU- accelerated	16.5	10.6
	Our method	8.8	7.9
Image size 512*480	Serial	290.6	88.7
	GPU- accelerated	54.2	34.0
	Our method	29.4s	25.1

According to the effect of the cooperative processing on CPU and GPU, the speedup factor of our algorithm can reach 9.9, and can performs at most 1.88 times faster than a traditional GPU-accelerated implementation.

We also designed an experiment to compare the computational consumption of different image sizes and different GPUs, where the FLOPS (FLoating-point Operations Per Second) of GPU2 is two times than GPU1, to show the salability of our algorithm, see in Fig. 6.

On one hand, when the experiments are done on the same GPU, the computational consumption increases the same times as the image size. On the other hand, the computational cost descends by half as the computational capability ascends by 2 times in the condition that the image sizes are the same. As a result, our CPU and GPU cooperative technique can process more sophisticated and greater amount of data on more advanced GPU.

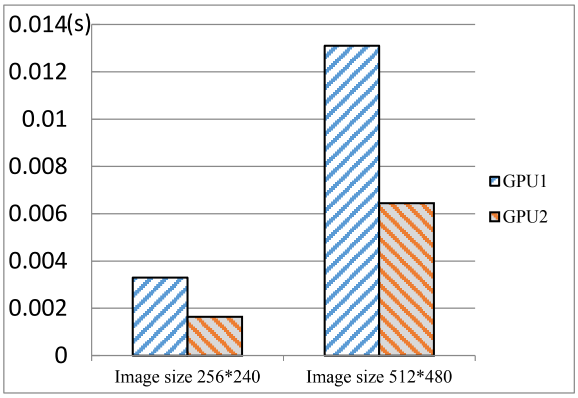

Fig. 6 GPU Computational Consumption

6 Conclusion

In this paper, we present a GPU and CPU cooperative accelerated road detection algorithm. The algorithm is robust against noise, shadows, and illumination variations. Meanwhile, the GPU & CPU cooperative parallel implementation makes sure that our method is real-time. The experiments verify that the algorithm is effective and real-time. Our cooperative technique can be used as reference for real-time system in intelligent navigation system.

7 Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant No.61272062, Fundamental Research Funds for the Central Universities of Hunan University under Grant No.531107040421 and Hunan Provincial Innovation Foundation for Postgraduate (CX2011B138).

8 References

[1] J. M. Álvarez, A. M. Ĺopez, "Road detection based on illuminant invariance," IEEE Trans. ITS, Vol.12, No.1, pp. 184-193, March 2011.

[2] Y. Wang, E. K. Teoh, D. Shen, "Lane detection and tracking using b-snake," Image and Vision Computing, Vol.22, No.4, pp.269-280, April 2004.

[3] H. Kong, J. Y. Audibert, J. Ponce, "General road detection from a single image," IEEE Trans. IP, Vol.19, No.8, pp.2211-2220, August 2010.

[4] Z. Tian, C. Xu, X. Wang, Z. Yang, "Non-parametric model for robust road recognition," 10th ICSP, Beijing, China, pp.869-872, October 2010.

[5] M. Foedisch, A. Takeuchi, "Adaptive real-time road detection using neural networks," Proc. 7th IEEE Int. conf. on ITS, Washington DC, United States of America, pp.167-172, October 2004.

[6] H. Xu, X. Wang, H. Huang, K. Wu, Q. Fang, "A fast and stable lane detection method based on b-spline curve," IEEE 10th Int. Conf. on CAID &CD, Wenzhou, China, pp.1036-1040, November 2009.

[7] J. M. Alvarez, T. Gevers, A. M. Lopez. "3d scene priors for road detection," 10th IEEE Conf. on CVPR, San Francisco, United States of America, pp.57-64, June 2010.

[8] R. O. Duda, P. E. Hart, D. G. Stork, "Multilayer neural networks," in Pattern Classification (Second Edition), pp.282-347, John Wiley & Sons, New York, 2001.

[9] J. Sun, C. Yang, "Curves and surfaces," in Computer graphics, pp.284-289, Tsinghua University Press, Beijing, 1995.

[10] T. Machida, T. Naito, "Gpu & cpu cooperative accelerated pedestrian and vehicle detection," ICCV 2011 Workshops. Barcelona, United States of America, pp.506-513, November 2011.