Adaptive Road Detection through Continuous Environment Learning

Mike Foedisch

Intelligent Systems Division
National Institute of
Standards and Technology
Gaithersburg, MD 20899
mike.foedisch@nist.gov

Aya Takeuchi

Advanced Technology
and Research, Corp.
Burtonsville, MD 20866
ayako.takeuchi@nist.gov

Abstract

The Intelligent Systems Division of the National Institute of Standards and Technology has been engaged for several years in developing real-time systems for autonomous driving. A road detection program is an essential part of the project. Previously we developed an adaptive road detection system based on color histograms using a neural network. This, however, still required human involvement during the initialization step. As a continuation of the project, we have expanded the system so that it can adapt to the new environment without any human intervention. This system updates the neural network continuously based on the road image structure. In order to reduce the possibility of misclassifying road and non-road, we have implemented an adaptive road feature acquisition method.

1. Introduction

The Intelligent Systems Division of the National Institute of Standards and Technology has been engaged for several years in developing real-time systems for autonomous driving. A road detection program is an essential part of the project. Algorithms for paved road detection [1, 3] have been extensively developed. Most are based on detecting road markings and lanes and cannot be applied to roads without any markings or lanes.

Compared to some approaches using Neural Networks [7] for autonomous driving [5], we use Neural Networks only for the road detection task by learning to differentiate the color distribution of road areas from other areas in the image. We have developed a real-time road detection application, which is independent of road markings and lanes [4]. During a short initialization step, feature data are automatically collected based on the typical appearance of road in images captured from the driver's point of view (road images). Then, a new Neural Network is trained and applied to future images. This procedure allows the system to detect road adaptively. However, it still required human involvement during the initialization step. As a continuation of the project, we have expanded the system so that it can adapt to the new environment without any human intervention. The system updates the neural network continuously based on the road image structure.

In this paper we first outline our previous approach briefly in section 2. In section 3, our new approach based on continuous learning is described in detail. Section 4 compares and discusses experimental results of the previous approach and two versions of the new approach.

2. Basic Adaptive Road Detection Approach using Neural Networks

Based on our previous work on adaptive road detection [4], we have developed the following basic approach using neural networks. The approach consists of two steps: the neural network training step and the road detection step.

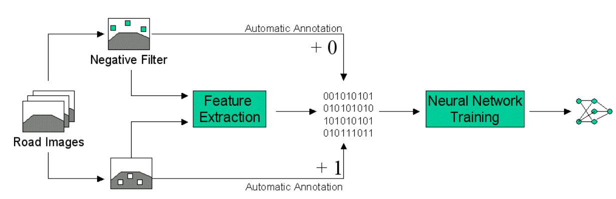


Figure 1: Overview of the Neural Network Training Step

Figure 1 gives an overview of the neural network training step. First, feature data are extracted from areas of the image defined by filters. Then, a neural network is trained on this features data using the filter type as classification label.

2.1. Feature Extraction

As introduced in [6], an “independent” color histogram consisting of 8 bins per channel is used [9]. This color histogram is computed for a 7-by-7-pixel window around each measuring point in the image. Additionally, we put the normalized x and y position [2] values of the current point of consideration into the set of features which results in a feature vector of 26 values.

2.2. Feature Filtering by Windows

The camera used as the sensor for detecting roads is positioned at the driver’s viewpoint. We take advantage of the fact that the road usually covers a trapezoidal area, which is centered in the lower part of the image.

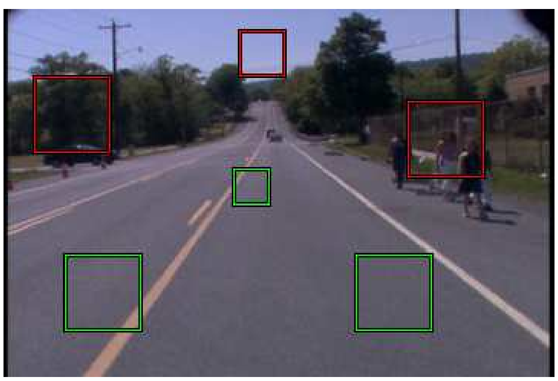


Figure 2: Example of three windows covering road area and another three covering non-road area

Based on the estimated road location in the image, feature vectors are collected from pre-defined windows, which cover either road (road windows) or non-road areas (non-road windows). Feature vectors extracted from the windows are automatically labeled as either road or non-road depending on the type of the window. The example in Figure 2 shows three windows placed in the road area of the image and three windows in non-road areas.

2.3. Neural Network Training

The current application uses a C++ based Neural Network library[1] [8]. The Neural Network receives 26 inputs (24 RGB histogram bins plus x and y coordinates) and consists of three layers. The first two layers contain four neurons each. The last layer is composed of one neuron, which generates the output. The Neural Network employs back-propagation learning [7].

2.4. Road Detection

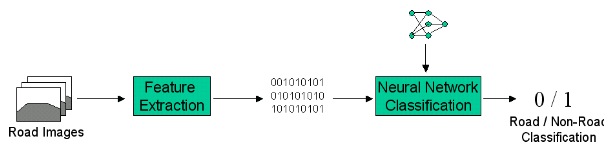


Figure 3: Overview of the Road Detection Step

Figure 3 outlines the road detection step. We overlay the input image with a raster of measuring points for which we extract feature vectors. Each feature vector is processed by the Neural Network, which was trained in the previous step. The resulting value is interpreted as either the road class or non-road class.

Figure 4: Sample Result

Figure 4 depicts a sample result where the area classified as road is drawn with white dots and the area classified as non-road shown with black dots.

---

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily best for the purpose.

We have enhanced the road detection accuracy by applying two post processing steps. First, noise is reduced by erosion and dilation methods. Second, we select only the largest detected region as the road area in the image.

3. Adaptability Approach through Continuous Learning

We previously developed a road detection system that relied on a Neural Network trained only in the beginning of the course. Although we could see reasonable results immediately after training the network, performance degraded as the environment and the road changed (e.g. due to chances in the sunlight) [4].

In order to solve this problem, we have added continuous update of the neural network. For this purpose, new feature data are constantly collected from the feature extraction windows. The re-training of the network currently takes about 600ms. In our implementation, we collect new feature data for about one second, which enables us to replace the neural network roughly every two seconds.

In this section, we describe two approaches of integrating continuous learning into a road detection system. The approaches differ in how the positioning of feature extraction windows is handled.

3.1. Fixed Windows Approach

Our first approach employing continuous learning for road detection follows the same basic method as the previous system described in Section 2. However, this version updates the neural network every two seconds.

This approach uses the same fixed windows position as the previous approach, which causes problems in certain traffic situations. Figure 5 is a camera view when the vehicle is turning a corner. In this situation, some windows violate our assumption. One of the road windows is placed completely in the non-road area, and one of the non-road windows covers both road and non-road content. When extracting feature data in this situation, the neural network has to handle contradicting information.

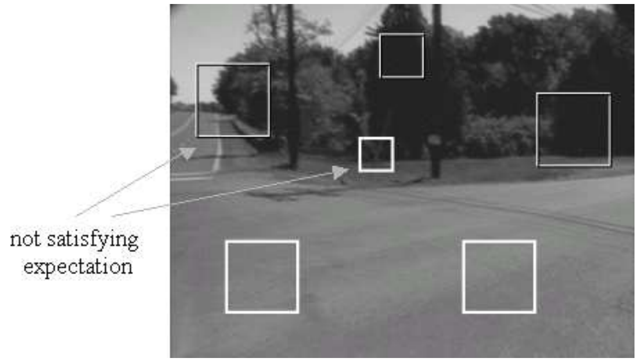


Figure 5: Example of windows violating the assumptions due to a change in the curvature of the road.

3.2. Dynamic Windows Approach

As mentioned in the previous section, the feature extraction from the fixed positioned windows causes problems in certain traffic situations (e.g. turning). To overcome this problem, we developed another approach in which the road windows automatically adjust to the current road shape.

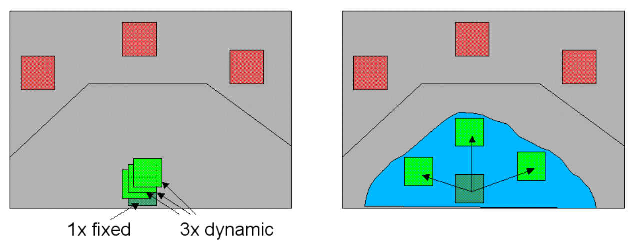


Figure 6: Approach for positioning dynamic windows

The algorithm for the placement of dynamic road windows is as follows:

The three non-road windows stay at their fixed position as before. On the other hand, instead of using three fixed road windows, we use four road windows. One window (reference window) is placed at a fixed position at the lower center of the image and the other three windows dynamically move from the reference window's location in pre-defined directions.

When the road detection system is started, all windows are placed in the image as depicted in Figure 6(left). From these locations, feature vectors are extracted and used for the initial neural network training. Because the initial classification result is based on the features extracted from the small area in the lower center of the image, the detected road area will appear as a mountain shape located around the reference window as shown Figure 6(right).

Based on these initial road detection results, the automatic moving of the three dynamic windows starts. Each window moves from the reference window's location in pre-defined directions; one window moves upward, another window moves up left and the other window moves up right. All of these windows move in each pre-defined direction as long as they fully contain road area, which was previously detected. From these new locations, feature vectors are collected and used to update the neural network.

This process lets the detected road area grow until it fully covers the actual road area in the image.

Switching Non-Road Windows

The problematic traffic situations described earlier not only affect the road windows but also the non-road windows. In order to reduce the effect of wrongly placed non-road windows, we manually switched off these windows whenever they covered road area.

4. Results

In order to analyze the improvement of our road detection's performance, we compared the results of the two new methods (fixed and dynamic windows approach) with the results of our previous system.

We have compared each algorithm's performance with manually annotated frames of video files. This allowed us to compute the false positive and false negative ratios. False positives refer to non-road areas in the image, which were classified by the system as road, while false negatives refer to road areas classified as non-road. We used the sum of both false positives and false negatives as an overall classification error calculated for each frame of the video sequence. After the error is calculated for each frame, we determined the minimum and maximum classification error throughout the whole video sequence and calculate the average as well.

While the overall classification error per frame allows us to compare the performance of several algorithms on the same frame, we can analyze the overall performance of each algorithm by comparing the minimum, maximum and average classification errors.

In the remaining section, we present sample results of the two new approaches and compare them with our previous approach. First, we want to point out two typical situations in which the algorithms perform differently. Second, we discuss the algorithms' performance on whole video sequences.

Abrupt Shadow Situation

Figure 7 shows the results for a situation in which shadow suddenly appears on the road. Two algorithms, the dynamic windows approach (*dwa*) and the previous approach (*pa*), handled the situation less accurately than the fixed windows approach (*fwa*).

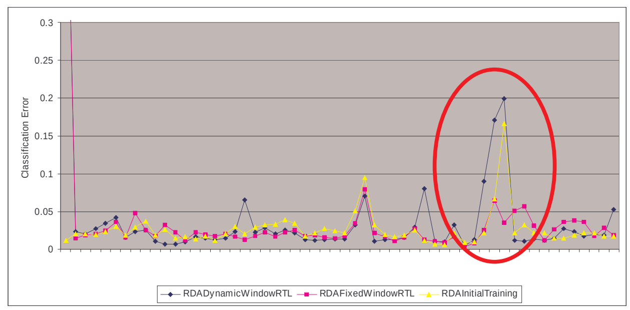


Figure 7: Classification error for abrupt shadow situation

The reason for the bad performance of *pa* is that there was no shadow in the beginning of the course and therefore the Neural Network was never trained on a shadow road. Likewise, the reason for *dwa's* performance is that it can't adapt to non-smooth changes. In contrast, the fixed road windows of *fwa* eventually (see arrow in Figure 8) covered the shadow area and allowed the Neural Network to be trained on it.

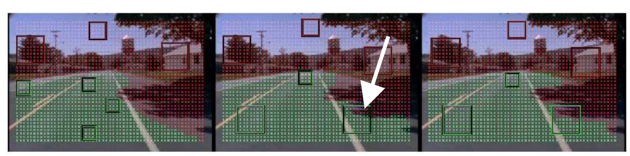


Figure 8: Abrupt Shadow Example for *dwa* (left), *fwa* (center) and *pa* (right)

Curvy Course Situation

Figure 9 shows the results for the situation of a curvy course. Both *dwa* and *pa* handled the situation more accurate than *fwa* did.

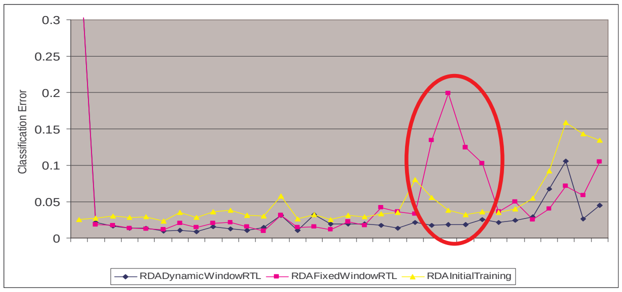


Figure 9: Classification error for a curvy course situation.

As the curvy road's appearance on the image differs from the road structure on which we based the location of the fixed windows, some of the windows violate our assumption. The contradictory feature data finally distorts the Neural Network. (see center in Figure 10)

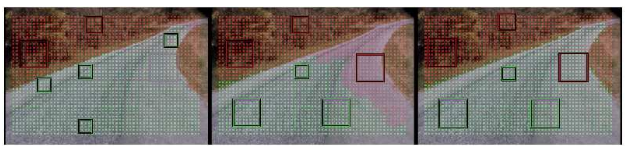


Figure 10: Curvy Course Example for *dwa* (left), *fwa* (center) and *pa* (right)

Based on the results of these two examples we can observe that each approach, *dwa* and *fwa*, has different advantages and disadvantages depending on the situation.

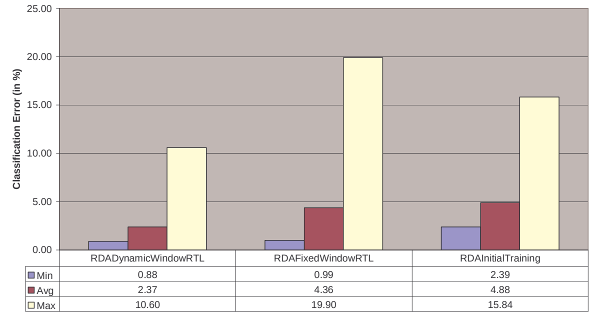
Next, we compare the algorithms in terms of their minimum, maximum and average classification errors.

Curvy Road

Figure 11 is a collection of snapshots of a video sequence with a fairly simple road but curvy course.

Figure 11: Sample Frames of the Curvy Road Example

Figure 12 depicts the classification errors for all three algorithms. Both of the new approaches show better results than *pa* in terms of the minimum and average classification error. The high peak of the maximum error for *fwa* is caused by a curvy course situation described earlier.



	RDADynamicWindowRTL	RDAFixedWindowRTL	RDAInitialTraining
Min	0.88	0.99	2.39
Avg	2.37	4.36	4.88
Max	10.60	19.90	15.84

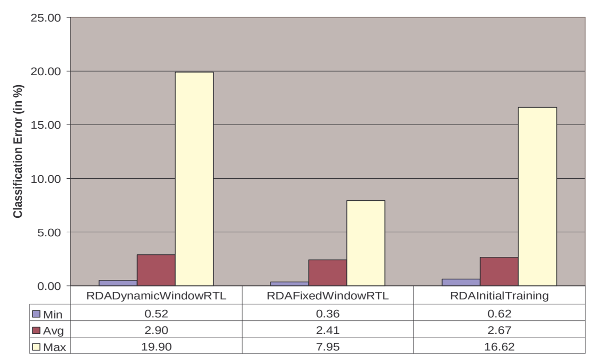
Figure 12: Classification Errors of *dwa* (left), *fwa* (center) and *pa* (right) for the Curvy Road Example

Straight Road

Figure 13 is a collection of snapshots of the video sequence showing a straight road. Most of the course appears similar, but some abrupt shadows occur during the course.

Figure 13: Sample Frames of the Straight Road Example

Figure 14 shows that *fwa* outperforms the other two algorithms. Both *dwa* and *pa* show a high maximum error, which is due to an abrupt shadow situation described earlier.



	RDADynamicWindowRTL	RDAFixedWindowRTL	RDAInitialTraining
Min	0.52	0.36	0.62
Avg	2.90	2.41	2.67
Max	19.90	7.95	16.62

Figure 14: Classification Errors of *dwa* (left), *fwa* (center) and *pa* (right) for the Straight Road Example

Shadow Road

Figure 15 is a collection of snapshots of a video sequence showing a simple road with shadow throughout the whole course.

Figure 15: Sample Frames of the Shadow Road Example

Compared to the classification errors of *pa* (see Figure 16), both of the new approaches show a slightly lower average classification error. However, we can see the best average performance for *dwa* and the lowest maximum error for *fwa*.

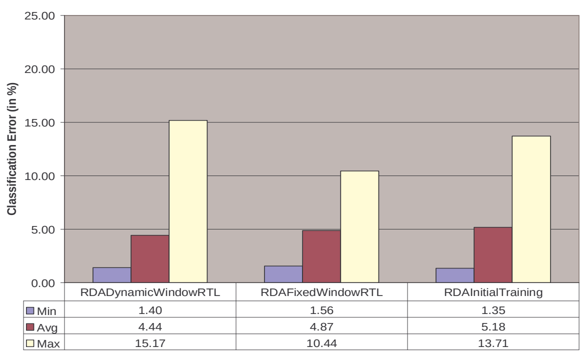


Figure 16: Classification Errors of *dwa* (left), *fwa* (center) and *pa* (right) for the Shadow Road Example

Our examples show for a variety of road types that at least one of the new algorithms performs better on average.

5. Conclusion and Future Work

We showed in this work that continuous learning does improve the performance of a road detection system. However, the key for an actual improvement lies in the method of positioning feature extraction windows. Our next steps will include higher level analysis of the type of road (e.g. two-lanes) and intersection (e.g. 4-way intersection) in order to control the positioning of windows based on the actual road situation.

Acknowledgments

Thanks to Christopher Rasmussen, who originally developed the feature extraction system and determined the methods for SVM and Neural Network generation, and Patrick Conrad, who enhanced the system by adding more features. We are also grateful for the support of the Army Research Laboratory and the Defense Advanced Research Projects Agency MARS program.

References

[1] M. Bertozzi, A. Broggi and A. Fascioli, "Vision-based Intelligent Vehicles: State of the Art and Perspectives," in *Robotics and Autonomous Systems* 32, 2000.

[2] P. Conrad and M. Foedisch, "Performance Evaluation of Color Based Road Detection Using Neural Nets and Support Vector Machines", in Proc. *Applied Imagery Pattern Recognition Workshop*, Washington D.C., 2003.

[3] E. Dickmanns, "Vehicles capable of dynamic vision", *Proc. Int. Joint Conf. On Artificial Intelligence*, pp. 1577-1592, 1997.

[4] M. Foedisch and A. Takeuchi, "Adaptive Real-Time Road Detection Using Neural Networks", to appear in *Proc. 7th Int. Conf. on Intelligent Transportation Systems*, Washington D.C., 2004.

[5] D. Pomerleau. "Neural Network Perception for Mobile Robot Guidance." PhD Thesis, Carnegie Mellon University, 1994.

[6] C. Rasmussen, "Combining Laser Range, Color, and Texture Cues for Autonomous Road Following", *in Proc. IEEE Inter. Conf. on Robotics & Automation*, Washington D.C., 2002.

[7] B. Ripley, "Pattern Recognition and Neural Networks", Cambridge University Press, 1996.

[8] A. Schmidt, "A modular neural Network architecture with additional generalization abilities for high dimensional input vectors" Master's thesis Manchester Metropolitan University, September 1996.

[9] M.J. Swain and D.H. Ballard, "Color Indexing", *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.