

Adaptive Road Following using Self-Supervised Learning and Reverse Optical Flow

David Lieb, Andrew Lookingbill, and Sebastian Thrun
Stanford Artificial Intelligence Laboratory
Stanford University
Gates Hall 1A; 355 Serra Mall; Stanford, CA 94305-9010
{dflieb,apml,thrun}@stanford.edu

Abstract—The majority of current image-based road following algorithms operate, at least in part, by assuming the presence of structural or visual cues unique to the roadway. As a result, these algorithms are poorly suited to the task of tracking unstructured roads typical in desert environments. In this paper, we propose a road following algorithm that operates in a self-supervised learning regime, allowing it to adapt to changing road conditions while making no assumptions about the general structure or appearance of the road surface. An application of optical flow techniques, paired with one-dimensional template matching, allows identification of regions in the current camera image that closely resemble the learned appearance of the road in the recent past. The algorithm assumes the vehicle lies on the road in order to form templates of the road’s appearance. A dynamic programming variant is then applied to optimize the 1-D template match results while enforcing a constraint on the maximum road curvature expected. Algorithm output images, as well as quantitative results, are presented for three distinct road types encountered in actual driving video acquired in the California Mojave Desert.

I. INTRODUCTION

The past few decades witnessed the emergence of numerous image-based techniques addressing various tasks critical to the development of robust autonomous driving systems for both on- and off-road conditions [1], [2]. The focus of much of this work has been the development of road following and lane tracking algorithms. In recent years, these technologies have received increasing publicity in both the civilian and military domains. Several automotive manufacturers are now offering lane-departure warning systems [3], [4], a first step in the realization of fully autonomous highway driving. The institution of the DARPA Grand Challenge [5], a competition between autonomous off-road vehicles in the Mojave Desert, has triggered extensive interest in camera-based road following algorithms.

Many road following algorithms are not adaptive. Some rely on *a priori* knowledge of specific visual characteristics of the road surface or structure, while others employ supervised learning techniques to learn to recognize a desired class of roads. For example, one class of algorithms searches for image edges that define the roadway, such as lane markers or road boundaries [6], [7]. Other methods exploit color cues unique to the road surface, often in combination with sophisticated segmentation algorithms [8] or known edge information [9]. Supervised learning algorithms also use these same types of

road cues to train classifiers which identify road regions [10], [11], [12]. Approaches of this nature are limited because they cannot adapt to changing road conditions without either re-tuning of *a priori* road identifiers or re-learning of trained classifiers with human supervision.

Some road following algorithms address this problem by incorporating adaptive learning techniques. Early work on adaptive algorithms used evolving templates consisting of traditional road cues [13] or color pixel clustering applied to known road models [14]. Other methods achieve adaptability by using color information of recent known road regions to search for future road regions [15] or by using color-based cues as the input to neural networks [16]. While these approaches successfully adapt to different or changing road types, each still relies on the presence of unique identifying features of the roadway, such as lane markings, edge boundaries, or distinct color or texture regions. Algorithms of this type would suffer on ill-structured roads lacking these distinct cues. Such roads possess neither clearly delimited boundaries nor unique surface features, and the color and texture of regions outside the roadway are often similar to those in the roadway, as in Fig. 1. Rasmussen presents an approach to handle this type of terrain wherein dominant texture orientations in each frame vote for the location of the road’s vanishing point [17]. This approach successfully computes road vanishing points on loosely defined roadways but it relies on texture artifacts left on the roadway by the passage of other vehicles. In some terrain types, such as desert, seasonal weather disturbances such as flash floods and wind storms may erase these texture artifacts.

We present an adaptive, self-supervised learning algorithm that targets this class of ill-structured roads using a reverse optical flow technique. Our algorithm makes no assumptions about the visual appearance of the roadway. Learning and adaptation are achieved according to a simple premise: use the region on which the vehicle currently lies as the definition of the roadway, and subsequently follow regions matching this description. The algorithm learns the most recent characteristics of the road by examining the appearance of the current vehicle location in a set of past camera images in which the vehicle was still some distance from its current location. Using reverse optical flow techniques, a set of templates is assembled from these previous images, each representing the

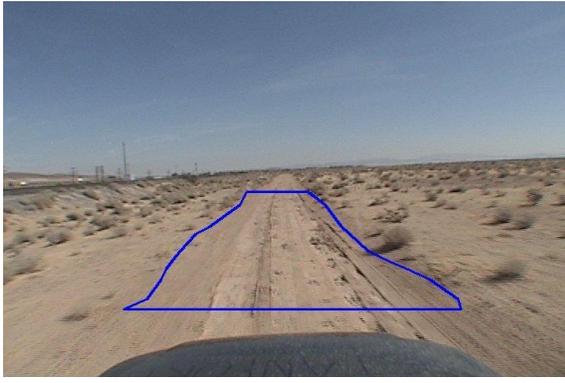


Fig. 1. An example of typical desert terrain containing a loosely defined road. The outline has been added by our algorithm, as described in the Methods section.

appearance of the most recent known road (the region currently directly in front of the vehicle) at a different distance. Template matching in the current camera image allows road localization without any assumptions about the visual characteristics of the roadway. This entire process is continually repeated, resulting in a self-supervised learning system able to adapt to changing road conditions. The results of running this algorithm on three sets of test videos taken from a moving vehicle traveling the 2004 DARPA Grand Challenge test course are included in the Results section of this paper.

Just as in [15], this algorithm makes the assumption that the vehicle is currently on the roadway in order to infer the current characteristics of the road. This learning and control system requires that the vehicle initially be driven at speed for a small time period to allow the storage of past images of the road. Any simple bootstrap algorithm or human supervision could be used to control the vehicle during the brief required startup period.

II. METHODS

There are three main components to the road following algorithm, which are described in the following subsections. First, a set of horizontal cross-sectional templates of the road at various distances is found using reverse optical flow techniques by assuming the region currently directly in front of the vehicle is drivable road. Second, these horizontal road templates are matched along horizontal lines at appropriate vertical heights in the current image, providing the locations in the image of regions that match the road's past appearance. Third, a dynamic programming technique is applied to the template matching responses along each horizontal line in order to find the globally optimal horizontal position of the road at each vertical height, subject to a constraint on the maximum possible curvature of the road. Finally, by interpolating between these optimal template matching positions and the template widths, road segmentation is achieved. Fig. 2 outlines the structure of the algorithm.

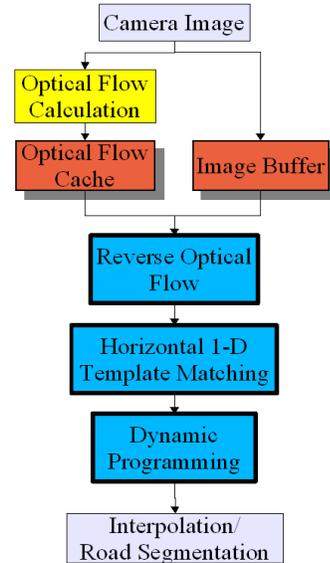


Fig. 2. Algorithm overview.

A. Finding Road Templates via Reverse Optical Flow

The algorithm assumes that the vehicle is initially traveling on the road and subsequently follows regions visually similar to the area directly in front of the vehicle, henceforth referred to as the *definition region*. The dark line in Fig. 4a shows the location of a typical one-pixel-high definition region used in our algorithm. To locate portions of the current image resembling this definition region via template matching, we wish to assemble a set of horizontal templates that reflect the characteristics of the definition region at various distances in front of the vehicle. Because perspective and illumination effects alter the width, brightness, and texture of the definition region at different distances, the best solution is to simply form the templates by directly pulling the current definition region from previous images when the region was further away.

To perform this reverse optical flow procedure, the optical flow fields between successive images must be computed for a sequence of frames prior to and including the current frame. For each pair of images, a set of unique features are found in the first image and traced to their locations in the subsequent image, with the displacement vectors constituting the optical flow field. In our implementation, features are first identified using the Shi-Tomasi algorithm [18], which selects unambiguous feature locations by finding regions in the image containing significant spatial image gradient in two orthogonal directions. Feature tracking is then achieved using a pyramidal implementation of the Lucas-Kanade tracker [19]. This approach forms image pyramids consisting of filtered and subsampled versions of the original images. The displacement vectors between the feature locations in the two images are found by iteratively maximizing a correlation measure over a small window, from the coarsest level up to the original level. The optical flow field between two consecutive images taken



Fig. 3. White lines depict an optical flow field between two consecutive images.

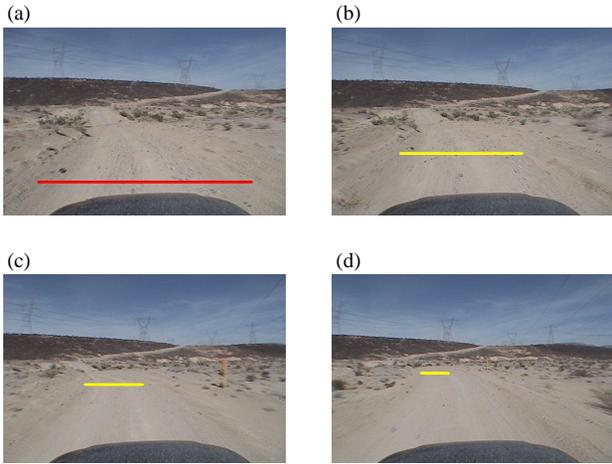


Fig. 4. (a) Dark line shows the definition region used in our algorithm. (b)-(d) White lines show the locations in previous frames to which reverse optical flow has traced the definition region.

from a data set acquired in the California Mojave Desert is shown by small lines in Fig. 3.

By dividing the optical flow field into a rectangular grid, it is possible to subsample and compress the optical flow field by storing only the mean displacement vector within each cell. In this way, the optical flow fields for a large number of frames can be easily stored and readily accessed in an array structure, with only slight loss of accuracy. Thus, any point in the current frame can be traced back to its origin in any prior frame whose optical flow has been cached by a simple summation of displacement vectors in a daisy-chain procedure.

This reverse optical flow procedure allows the location of the definition region in previous frames to be found with good accuracy. Sampling the traced-back definition region in a set of frames progressively further in the past then provides a set of horizontal templates of the definition region at various distances. Fig. 4(b-d) show the results of the reverse optical flow procedure applied to the definition region shown in Fig 4a. Horizontal templates such as those located along the white lines in Fig. 4(b-d) then serve as cross-sectional templates used to locate the road in the current image.

B. Horizontal 1-D Template Matching

Armed with a set of horizontal templates that depict the appearance of the definition region at various distances, standard template matching algorithms can be used to search for the most likely position of the road at various vertical heights in the current image. To ensure that the road in the current image is roughly the same width as the horizontal templates, these vertical search heights are chosen as the same vertical heights from which the definition region templates were drawn, with one caveat. Changes in scene topology and vehicle pitch can drastically alter the distance of a particular cross-section of road as a function of its vertical position in the image. To mitigate this effect, a simple Hough transform-based horizon detector is used to scale the vertical heights of the template search lines according to the vertical height of the horizon in the current image.

Because both the templates and the search space are horizontal slices, templates taken from curved roads appear and behave almost exactly as those from straight road segments. The only effect is that horizontal templates taken from roads with different orientation than the vehicle's current path will be artificially wide, as the horizontal cross-section of the road is wider at these points. This is the same effect that would be produced if the vehicle was undergoing moderate amounts of roll. The template matching measure, combined with the dynamic programming procedure described in the next section, serves to alleviate problems caused by these effects. In the presence of significant roll, however, problems with the horizon detector could adversely affect the accuracy of the algorithm.

An SSD (sum of squared differences) matching measure is used to compute the strength of the template match along each horizontal search line. The normalized SSD measure is defined as follows (where I is the image, T is the template, $\{x', y'\}$ range over the template, and $\{x, y\}$ range over the image):

$$R(x, y) = \frac{\sum_{x'} \sum_{y'} [T(x', y') - I(x + x', y + y')]^2}{[\sum_{x'} \sum_{y'} T(x', y')^2 \cdot \sum_{x'} \sum_{y'} I(x + x', y + y')^2]^{0.5}} \quad (1)$$

Since the search space for each template is a single horizontal line, and the template height is small (typically around 10-20 pixels), this matching measure can be quickly computed. Fig. 5a shows a visualization of the matching response for a set of 10 horizontal templates along 10 horizontal lines in a typical camera image (shown in Fig. 5b), with the road curving to the left in the distance. White regions indicate a strong match while dark regions indicate a poor match. Although the matching is performed along only a single horizontal line, the responses in the figure have been widened vertically for visibility. Clearly, strong responses occur in image regions near the center of the road. However, it is also evident that strong responses may also occur elsewhere along each search line if the road is not clearly distinguishable from the rest of the

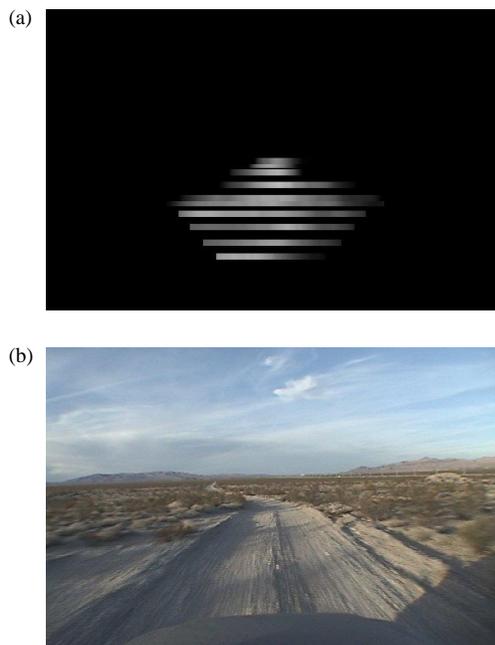


Fig. 5. (a) Visualization of SSD matching response for 10 horizontal templates. White indicates a strong response, black a weak response. (b) Corresponding input frame.

scene, as is the case in the lower left portions of Fig. 5a and 5b, where the shadows and lack of vegetation combine to make some template matches to the left of the road appear attractive. This collection of SSD matching responses can also be used as a confidence measure. If the value of the best SSD measure drops sharply for all of the horizontal lines in the current image at once, it is likely that the vehicle has left the road or that the characteristics of the road have changed drastically. Actions could be taken at this point such as alerting a human operator or beginning an active search for areas with the characteristics of the road last seen.

Fig. 6 indicates the position of the maximum SSD response along each horizontal search line with dark circles. While several of the circles correctly locate the position of the road, it is clear that the maximum response location is not necessarily correct. Choosing the location of maximum response along each line would also allow physically unrealizable estimates of the position of the road at various distances.

C. Dynamic Programming for Road Location Optimization

The problem of finding the globally optimal set of road location estimates while satisfying a constraint on the maximum curvature of the road lends itself well to the use of dynamic programming. Dynamic programming variants have been used in the past for both aerial [20] and ground-based [21], [22], [23] road and lane detection. The goal of the dynamic programming module is to determine the horizontal position of each template along the horizontal search lines, such that when taken together the positions minimize a global cost function. The cost function used in this algorithm is simply the arithmetic inverse of the SSD response at a

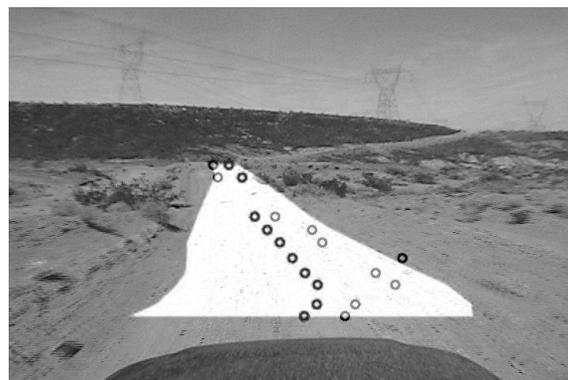


Fig. 6. Dark circles represent locations of maximum SSD response along each horizontal search line. Light circles are the output of the dynamic programming routine. The white region is the algorithm's final output and is interpolated from the dynamic programming output and template widths.

particular horizontal position along each line, summed over all search lines. Dynamic programming is then performed as usual: The horizontal search lines are processed from the topmost downward, with the cost at each horizontal position computed as the SSD cost at that particular location plus the minimum cost within a limited window around the current horizontal position in the search line above. The horizontal position of this minimum cost location is also stored as a link. The window restriction serves to enforce a constraint on the maximum allowable curvature of the road as well as to reduce the computation time of the optimization. Once the bottommost search line has been processed in this manner, the globally optimal solution is found by following the set of stored links that point to the minimum cost position in the search line above. The path traversed represents the center line of the road estimate.

The output of the dynamic programming algorithm is shown as light circles in Fig. 6. The entire road can now be segmented by interpolating between the optimal positions determined by dynamic programming. The white region in Fig. 6 illustrates road segmentation by interpolation using a 4th-degree polynomial fit of the dynamic programming output. The width of the road region is linearly interpolated from the widths of the horizontal templates.

III. RESULTS

A. Test Data

Road detection by one-dimensional template matching using reverse optical flow worked well in a variety of test conditions. Single frame results from three different 720 x 480 pixel video sequences taken in the Mojave Desert are shown in Fig. 7. Each column of Fig. 7 contains results from one of these data sets. The first video sequence consists mainly of a straight dirt road in a sparse desert environment and illustrates the ability of our algorithm to correctly find the road even in environments containing many regions visually similar to the road surface. The second video sequence consists of a gravel road winding up and down a rocky hill in broad daylight. The

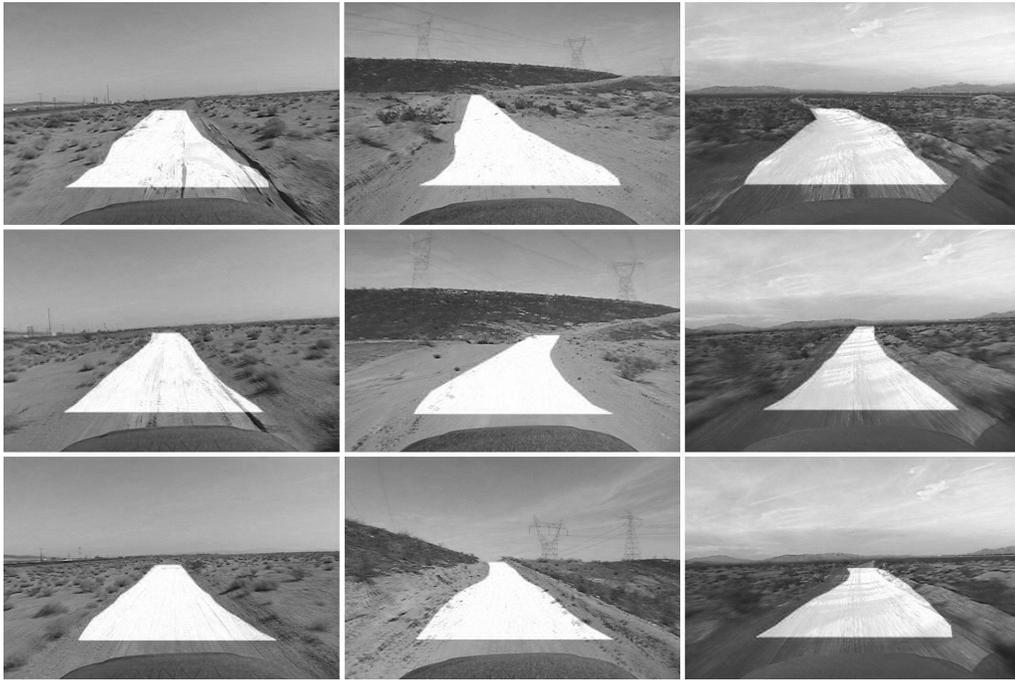


Fig. 7. Single frame algorithm output for three Mojave Desert data sets. Each column contains results from one of the three video sequences.

third sequence consists of a curved road traversed late in the day when shadows intermittently fall on the roadway.

The results presented here were found by 1-D template matching of a set of 10 horizontal templates acquired using the reverse optical flow procedure. The templates are samples of a definition region directly ahead of, and slightly wider than, the vehicle. They were taken from different times in the past, ranging from 1 frame to roughly 200 frames prior to the current frame. The particular temporal samples were chosen to provide an evenly spaced set of templates. Each template is 20 pixels high, and the definition region and templates were refreshed every 10 frames to accommodate gradual changes in road appearance. Optical flow fields were measured using a set of 3000 feature correspondences and cached into a grid structure of 96 rectangular cells covering the entire camera image. Interpolation of the dynamic programming output was achieved using a 4th-degree polynomial fit.

B. Quantitative Test Metrics

To quantify the overall performance of the algorithm, we have evaluated the results of the three 1000-frame Mojave Desert data sets described above using two performance metrics. For our own comparison purposes we have also implemented and tested two additional road following algorithms: one color-based and one texture-based. The color-based algorithm labels pixels with color values within a tolerance range of a target color acquired from the definition region, as shown in Fig. 8. The texture-based algorithm labels image regions displaying texture similar to that of an image patch in the definition region, as shown in Fig. 9 (texture matches above the horizon were ignored). Both algorithms were manually

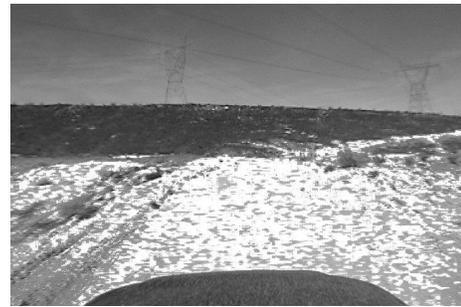


Fig. 8. White pixels represent the output of the color-based road following algorithm

tuned to optimize correct pixel coverage. Neither comparison algorithm performs any reverse optical flow. While these two algorithms are somewhat elementary, they serve to illustrate several of the advantages of the algorithm described in this paper. However, since our implementations of these approaches were naive, the fact that our algorithm quantitatively outperforms them is not informative. The outputs of these algorithms are shown in our comparison videos, but are not included in the Results section. The two performance metrics used to evaluate our algorithm are described below.

Pixel Coverage Metric: The first metric compares pixel overlap between the algorithm output and ground truth images in which the road has been segmented by a human operator, as shown in Fig. 10. The number of pixels in the frame that have been incorrectly labeled as roadway is subtracted from the number of correctly labeled roadway pixels. This number is then divided by the total number of pixels labeled

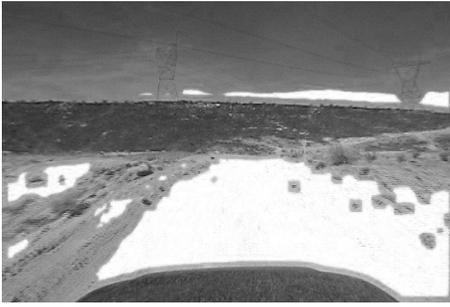


Fig. 9. White pixels represent the output of the texture-based road following algorithm



Fig. 10. Typical human-labeled ground-truth image

as road by the human operator for that frame. Using the metric proposed here, a score of 1.0 would correspond to correctly identifying all the road pixels as lying in the roadway, while not labeling any pixels outside the roadway as road pixels. A score of 0.0 would occur when the number of actual road pixels labeled as roadway is equal to the number of non-roadway pixels incorrectly identified as being in the road. If more pixels were incorrectly labeled as roadway than actual road pixels correctly identified, negative scores would result. This measure is computed once per frame and averaged over the entire video sequence. While this pixel coverage metric is easily visualized and simple to compute, it must be recognized that, due to perspective effects, it is strongly weighted towards regions close to the vehicle.

Line Coverage Metric: The second metric alleviates the distance-related bias by comparing pixel overlap separately along a set of horizontal lines in the images. Specifically, five evenly spaced horizontal lines are chosen ranging in vertical position between the road vanishing point and the vehicle hood in the ground-truth image. Success scores are calculated just as in the first metric, except they are reported individually for each of the five lines. The metric returns five sets of success scores computed once per frame and averaged over the entire video sequence.

C. Performance Metric Results

Fig. 11 shows the performance of the algorithm proposed in this paper on the three different video sequences, evaluated using the pixel coverage metric. The performance of the color and texture algorithms is drastically lowered in all three test

	Video 1: straight road	Video 2: curved hill	Video 3: curved with shadows
Metric Score	0.6958	0.6983	0.6322

Fig. 11. Pixel coverage results on the three test video sequences

sequences due to large rates of incorrectly labeled non-road pixels. This is in keeping with the large percentage of area outside the roadway in these videos that displays similar color and texture characteristics to regions on the roadway.

The strengths of the proposed algorithm are best highlighted in the first test video sequence, which consists mainly of a straight dirt road surrounded by large non-road regions with visual characteristics strikingly similar to that of the road. Despite the fact that the road is only loosely defined with respect to the surrounding regions, our algorithm is able to correctly locate the roadway with a strong pixel coverage metric score. Our color- and texture-based algorithms do poorly in terrain of this sort.

The second test video sequence, which contains a gravel road curving up a hill, presents fewer problems for all three algorithms. The increased presence of desert vegetation outside the roadway helped to reduce the false positive rates of the color and texture approaches. Curved roads with significant elevation changes do not seem to adversely affect our algorithm, as compared to the straight road found in the first test sequence.

The addition of intermittent shadows as found in the third test sequence does slightly affect the performance of our algorithm. This effect has an intuitive explanation and is discussed in the following section. It is interesting to note the degree to which the added presence of shadows hinders each algorithm relative to its performance in the similar, shadow-free, environment of the second video.

Fig. 12 shows the performance of the algorithm on the same three data sets, now evaluated using the line coverage metric. Metric scores are graphed for a set of five evaluation lines increasingly distant from the vehicle. As could be expected, the performance of the algorithm generally declines as the distance from the vehicle increases. The proposed algorithm achieves very low false positive rates by making no assumptions about the general appearance of the road and following regions that adhere to its learned roadway information. The inability of our algorithm to achieve high rates of correct roadway labeling near the horizon is at least in part due to the fact that optical flow records are only stored for a fixed number of frames in the past. Therefore the definition region can never be traced back all the way to the horizon.

Videos of the three 1000-frame test sets, showing the results of tracking with the proposed algorithm as well as the simple color and texture algorithms, are available at http://www.visiondemo.net/road_following/. The algorithm runs at 3Hz on a 3.2GHz PC at 720 x 480 pixel resolution.

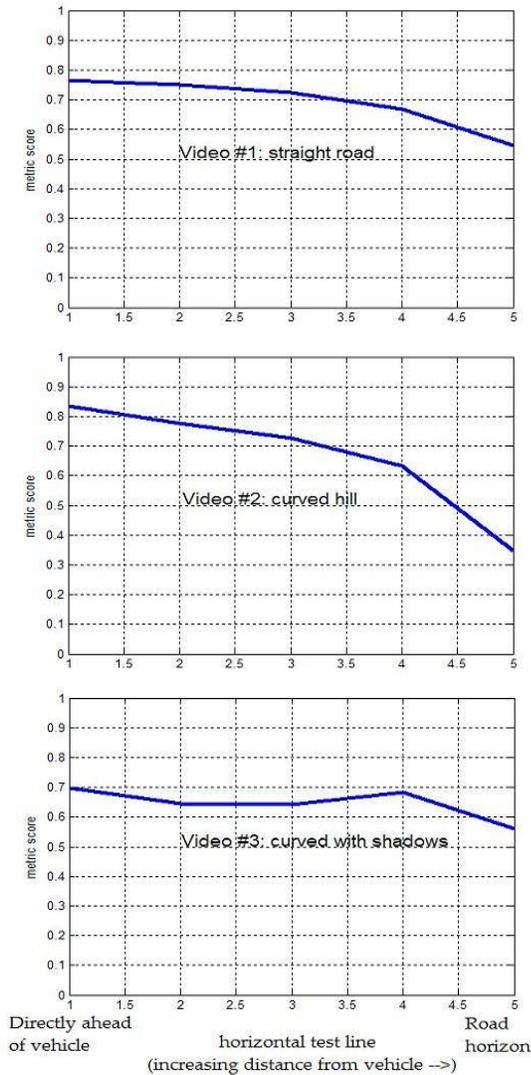


Fig. 12. Pixel coverage results are shown at different distances from the front of the vehicle towards the horizon for the three video sequences.

D. Assumptions and Limitations

The road following algorithm described here performs well in the types of environments depicted in the three Mojave data sets. The algorithm has been designed to follow typical desert roads, though, and therefore some limiting assumptions have been made. The main concept of the algorithm is to locate the road in front of the vehicle by searching for regions similar in appearance to what the road was known to look like in the (not too distant) past. As a result, we are forced to assume that the general appearance of the road surface will not change instantaneously. While this is usually a fairly safe assumption, we see from the third data set that intermittent shadows or other lighting variations violate this assumption and adversely affect the algorithm’s performance. Fig. 7 illustrates two instances of this assumption breaking down. The top left panel shows the algorithm attempting to

avoid a portion of the road containing tire ruts left in the mud by another vehicle. In the bottom right panel, the algorithm is biased toward the right side of the road due to the presence of shadows on the left. It is interesting to note, though, that the shadows themselves do not create this bias, as the algorithm has no difficulties on other portions of the shadow-filled test video. The effect is entirely dependent on the presence or lack of shadows in the set of horizontal templates. The templates in use in the bottom right panel happened to be chosen along lines predominantly free of shadow.

In this implementation, we have also chosen the width of the definition region to be fixed at slightly larger than the width of the vehicle without regard to the road’s actual width. This is consistent with the roads found in our data sets, but had the width of the road been different or had the road gradually changed width along the way, the algorithm would have had no way of compensating. Similarly, the particular prior frames from which the templates are drawn have been hard-coded, therefore fluctuations in vehicle speed affect the range of the road prediction. When the vehicle slows down, the furthest template moves closer to the vehicle and the segmented road shortens; when the vehicle speeds up, the templates extend farther from the vehicle and the segmented road effectively lengthens. These frame choices could be automatically tuned to current vehicle speed, though no such approach has been implemented in the algorithm presented here.

Another critical requirement of the algorithm is its ability to find and track image features in the video sequence. Knowledge of the road’s appearance relies entirely on the ability of the reverse optical flow procedure to accurately locate the definition region in past images. Low contrast image sequences resulting from poor lighting or camera saturation typically lack sufficiently unique image features. Also, regions lacking physical texture, such as smooth, homogeneous desert ground, also present problems for feature identification and tracking. On reasonably planar terrain, though, the optical flow field in these smooth regions can be interpolated from surrounding feature-rich regions. The number of features tracked per frame is proportional (though with diminishing returns) to algorithm performance, as a better estimate of each frame’s optical flow is achieved from a larger sample of flow vectors.

IV. CONCLUSION

We have proposed an adaptive, self-supervised learning algorithm for the detection of unstructured desert roads from a vehicle-mounted camera that relies on no assumptions about particular characteristics of the roadway. The region directly in front of the vehicle is assumed to lie in the roadway, and this region is then identified in a set of past camera images using a reverse optical flow routine. A set of horizontal templates are assembled from the location of this region in the past images, which are then matched, at similar vertical heights, along horizontal lines in the current image. The SSD match responses are then fed into a dynamic programming routine that determines the globally optimal estimate of the location of the road at these different heights in the image, given

the physical constraints on the possible radius of curvature of the road. Interpolation between the output positions of the dynamic programming routine provides the centerline of the road estimate, and interpolation between the horizontal template widths determines the segmented road width.

The algorithm was tested on three video sequences containing varying desert road conditions. Two separate metrics were used to gauge the success of the proposed algorithm, as compared to human-labeled ground-truth. The algorithm has been shown to perform well in the challenging environments for which it was devised.

ACKNOWLEDGMENTS

This research has been financially supported through the DARPA LAGR program. The views and conclusions contained in this document are those of the authors, and should not be interpreted as necessarily representing policies or endorsements of the US Government or any of the sponsoring agencies.

REFERENCES

- [1] M Bertozzi, A Broggi, A Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," in *Journal of Robotics and Autonomous Systems*, 2000. pp. 1-16.
- [2] Guilherme N. DeSouza and Avinash C. Kak, "Vision for Mobile Robot Navigation: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 2, February 2002. pp. 237-267.
- [3] T. Pilutti and A. G. Ulsoy, "Decision Making for Road Departure Warning Systems," in *Proceedings of the American Control Conference*, June 1998.
- [4] S. Plumb, "Lane-keeping System put to the Test," *Autotech Daily*, May 17, 2004. p. 2.
- [5] Defense Advanced Research Projects Agency (DARPA), DARPA Grand Challenge, Online source: <http://www.darpa.mil/grandchallenge>.
- [6] C. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.
- [7] A. Takahashi, Y. Ninomiya, M. Ohta, and K. Tange, "A Robust Lane Detection using Real-time Voting Processor." In *Proc. IEEE ITS*, 1999.
- [8] L.J. Liu, M.W. Ren, Y. L. Cao, and J. Yang, "Color road segmentation for ALV using pyramid architecture," in *Proc. SPIE Vol. 2028*, p. 396-404, *Applications of Digital Image Processing XVI*.
- [9] Y. He, H. Wang, and B. Zhang, "Color-Based Road Detection in Urban Traffic Scenes," in *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, No. 4, December 2004. pp. 309-318.
- [10] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc. Int. Conf. Robotics and Automation*, 2002.
- [11] D. Kuan, G. Phipps, A.-C. Hsueh, "Autonomous Robotic Vehicle Following." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Sep 1998, pp.648-658.
- [12] D. A. Pomerleau, *ALVINN: an autonomous land vehicle in a neural network*, *Advances in neural information processing systems 1*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1989.
- [13] D. A. Pomerleau, "RALPH: Rapidly Adapting Lateral Position Handler," in *IEEE Symposium on Intelligent Vehicles*, 1995.
- [14] J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. Int. Conf. Robotics & Automation*, 1991, pp. 2496-2501.
- [15] J. Fernandez and A. Casals, "Autonomous Navigation in Ill-Structured Outdoor Environments," in *Proc. Int. Conf. Intelligent Robots and Systems*, 1997.
- [16] M. Foedisch and A. Takeuchi, "Adaptive Real-Time Road Detection Using Neural Networks," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, 2004.
- [17] C. Rasmussen, "Grouping Dominant Orientations for Ill-Structured Road Following." In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004*.
- [18] J. Shi and C. Tomasi. "Good Features to Track." *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 593-600, 1994.
- [19] J. Bouguet. "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm." Intel Corporation, Microprocessor Research Labs, 2000. OpenCV Documents.
- [20] A. P. Dal Poz, G. M. do Vale, "Dynamic programming approach for semi-automated road extraction from medium- and high-resolution images," *ISPRS Archives*, Vol. XXXIV, Part 3/W8, Munich, 17-19. Sept. 2003.
- [21] K. Redmill, S. Upadhyaya, A. Krishnamurthy, U. Ozguner, "A Lane Tracking System for Intelligent Vehicle Applications," in *Proc. IEEE Intelligent Transportation Systems Conference*, 2001.
- [22] H. Kim, S. Hong, T. Oh, J. Lee, "High Speed Road Boundary Detection with CNN-Based Dynamic Programming," in *Advances in Multimedia Information Processing - PCM 2002: Third IEEE Pacific Rim Conference on Multimedia Hsinchu, Taiwan, December 16-18, 2002*. pp. 806-813.
- [23] Dong-Joong Kang, Mun-Ho Jung, "Road lane segmentation using dynamic programming for active safety vehicles," *Elsevier Pattern Recognition Letters* 24, 2003.