ADAPTIVE ROAD IMAGE SEGMENTATION FROM LADAR-DERIVED LABELS

Christopher Rasmussen, William Ulrich

University of Delaware
Department of Computer & Information Sciences
Newark, DE

ABSTRACT

We present an approach to image-based road segmentation for autonomous driving in which an appearance model is adaptively learned from laser range-finder data. By tracking linear configurations of ladar obstacles as putative road edges and backprojecting into the image, a coarse partition of pixels into high-confidence on-road and off-road regions, as well as unlabeled bands of uncertainty between them, is obtained. A model of the current appearance of the road is learned by running a classifier on labeled image features. The immediate effect is a more refined segmentation at the pixel level indicating nonlinear shape features such as curves, dips, and rises; and some inference of the road geometry beyond the ladar range. At a higher level, the proposed image-ladar interaction offers an approach to segmenting novel roads and in changing illumination conditions without manual intervention. Some results using support vector machines and neural networks as the classifiers on a varied set of desert road images are discussed.

1. INTRODUCTION

The DARPA Grand Challenge robot races in 2004 and 2005 drew considerable attention in the vision and image processing community to algorithms for autonomously following unpaved paths and roads.

Successful strategies for image-only road segmentation have tended to cluster based on certain assumptions about the characteristics of the road scene. For example, edge-based methods such as those described in [1, 2, 3] are often used to identify lane lines or road borders, which are fit to a model of the road curvature, width, and so on. These algorithms typically work best on well-engineered roads such as highways which are paved and/or painted, resulting in a wealth of high-contrast contours suited for edge detection.

An alternative set of methods for road tracking is region-based [3, 4, 5, 6]. These approaches use appearance characteristics such as color or texture measured over local neighborhoods in order to formulate and threshold on a likelihood that pixels belong to the road area vs. the background. These features are non-geometric–that is, image location is not considered for segmentation. When there is a good contrast for the cue chosen, there is no need for the presence of sharp or unbroken edges, which tends to make these methods more appropriate for unpaved rural roads.

Fig. 1. Vehicle used to capture data for this paper. The ladar is mounted on the bumper (indicated by the red box) and the camera is one of the stereo pair over the windshield (the blue box).

In this paper we present a machine learning approach to road segmentation that combines elements of both groups. The primary novelty of the algorithm is in its derivation of pixel class labels online from a calibrated laser range-finding sensor (aka *ladar*). One weakness of classification-based approaches to road segmentation has typically been the question of where the labeled examples of road and non-road image patches to train the classifier come from. Labels can be obtained in two general ways. The first is through manual annotation of a set of images that are representative of the roads to be driven [7]. Aside from the labor involved in doing this, there is a problem of generality: the data set used for training needs to capture all possible visual variations due to time of day, weather, and season; different road materials such as sand, gravel, and asphalt; and the camera's photometric properties. Adaptive algorithms attempt to adjust the appearance model over time [8, 9], but to do so they must make an assumption about what part of the image

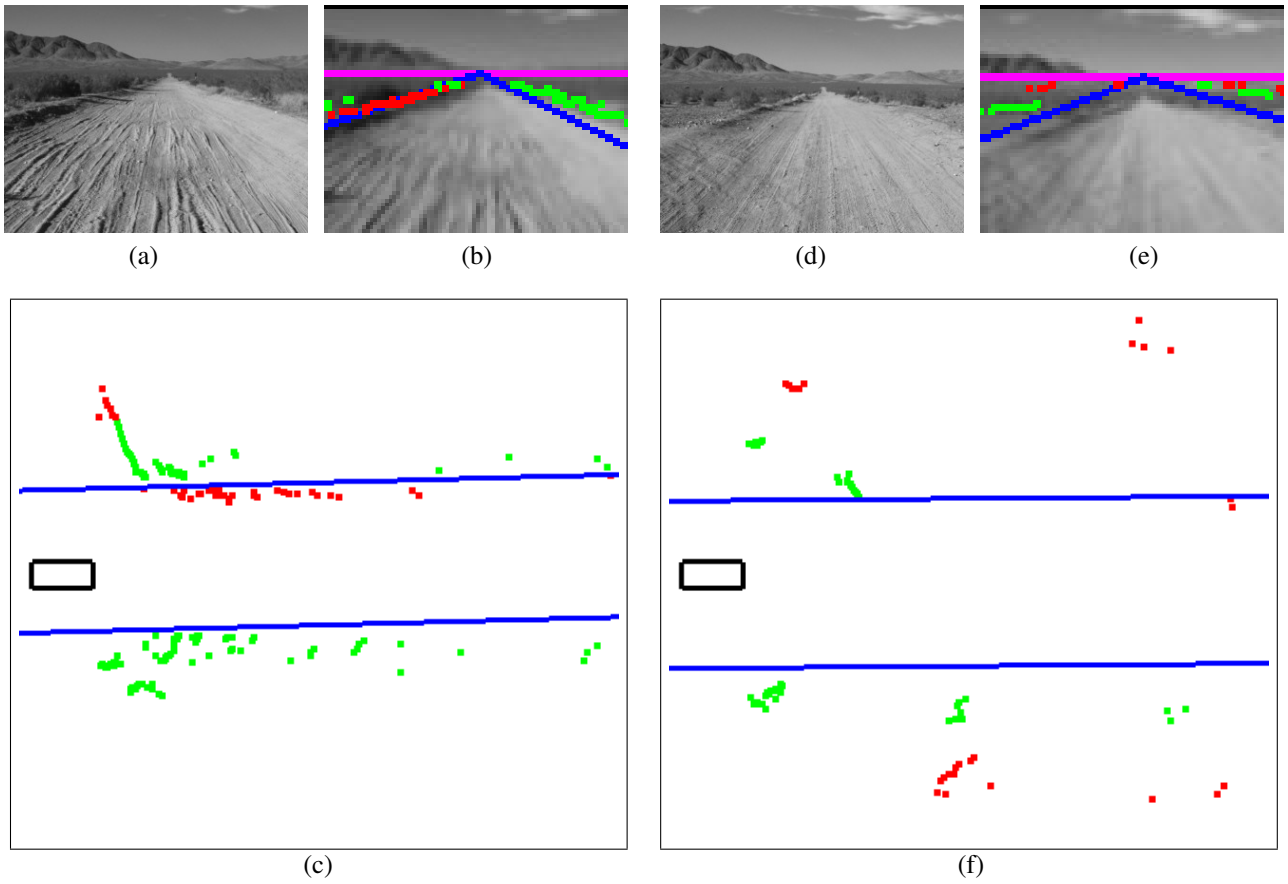


Fig. 2. Sample camera images and ladar range values for two road scenes. (a) 320×240 high-resolution image (not used for image processing); (b) 80×60 low-resolution version with ladar range values and fitted lines projected (horizon and road vanishing point are derived from image only); (c) Bumper ladar range values and fitted road edges in vehicle coordinates. (d), (e), and (f) are same for scene with sparser obstacles. Outliers in the fitting process are drawn in red, inliers in green.

is road and what is not, which may compound errors gradually. Too conservative an assumption underutilizes available information.

Here we suggest a principled approach to automatically labeling the majority of a road image's pixels based on the structural cues provided by a ladar. These labels are then used to learn a model of the road's current visual appearance which can be applied to the remaining unlabeled pixels, resulting in a more accurate segmentation than either sensor alone can easily extract. Visual and structural modalities are clearly complementary in many situations: vision alone may be inadequate or unreliable in the presence of strong shadows, glare, or poor weather, but bushes, fences, ditches, or curbs along the road can serve as a guide. Conversely, road boundaries do not necessarily coincide with 3-D structures–the height border between a dirt road and short grass, for example, is undetectable by most current methods and sensors, but the color contrast between the two is unmistakable.

After this paper was submitted, we learned that a somewhat similar approach was developed independently for Stanford's 2005 DGC team and described in a paper [10] published simultaneously with this one. The classification techniques described in that paper differ from ours in a number of ways. For example, they did not rely on negative examples as we do, they do not detail how their region of positive examples is derived from the ladar cost map, and they use color as a feature while we only have grayscale images. Another important point of departure is our strong assumption that a road is always visible. We regret that there is not enough space or time to include a more detailed comparison of the two algorithms in this paper.

In the following sections we will detail the methods by which the ladar range data is analyzed and transformed to coarsely indicate the road region in the image. We then examine how this is used to guide classifier training and subsequent prediction of unlabeled pixels' classes, and then present results demonstrating the accuracy of the system

over a range of different road image situations. Finally, we talk about ongoing work to improve and extend the system.

2. METHODS

2.1. Sensor configuration

A single SICK LMS ladar mounted on the vehicle bumper with a scan plane parallel to the ground was used here. The camera was mounted above the vehicle's windshield, pitched down slightly. The sensor configuration is pictured in Figure 1. The ladar is about 0.5 m above the ground, so it only "sees" obstacles of that height or greater. As the examples will show, there are certainly some shorter obstacles that are not detected in the data sets we used. Of course this configuration is also blind to so-called *negative obstacles* or holes/ditches, but this could easily be remedied with another ladar mounted with a downward pitch.

A sample high-resolution (320×240) grayscale image from the camera is shown in Figure 2(a). A lower-resolution (80×60) version of the same image is shown in Figure 2(b). The distribution of ladar range measurements $\mathbf{z}$ for the scene pictured is drawn below in Figure 2(c) as green and red points (the colors are explained below). The projection of these points into the image is shown in Figure 2(b). Analogous views are shown for a different scene in Figure 2(d-f). The primary difference between the scenes is that for the one on the left the ladar points are grouped in two clear linear clusters, while in the right-hand scene the ladar points are more scattered because the off-road vegetation is sparser. We assume here that most of the road scenes are like the one on the left.

2.2. Ladar line fitting

The method we use to extract the road region from the ladar range values is based on the RANSAC method [11] for robust line fitting. Intuitively, we want to fit lines to the roughly linear clusters of ladar range values that we presume demarcate the left and right road edges. Assuming that the vehicle is headed more-or-less down the road, we can divide the ladar points into *left* and *right* sets based on which side of the ladar midline (i.e., straight ahead) they are on. We need a robust fitting procedure because even when ladar points are clustered linearly, there are often outliers in the form of vegetation and other structure further away from and not associated with the road.

Constraining the left and right lines to be parallel as they ought to be instead of fitting them independently, we need a sample of three points to fully parametrize a linear road segment–two from one side and one from the other. The standard criterion for a point $\mathbf{p}$ to be an inlier is for its unsigned distance to the line to be less than a threshold τ. We modify this criterion because we do not want each fitted line to go through the middle of a linear obstacle cluster. Rather, to be conservative we want it to be on the *inside* of the cluster with respect to the road. Thus, we use the signed point-line distance to determine which side of the line $\mathbf{p}$ is on (depending on the line under consideration and whether $\mathbf{p}$ is in *left* or *right*). After much experimentation, we settled on the criteria that *any* point on the inside of the line is regarded as an outlier and outside points are only inliers when their distance is below $\tau = 5$ meters. Inliers are drawn in green and outliers in red in Figures 2(c) and (f).

A priori highly unlikely configurations can be eliminated by not considering samples that lead to road segments with widths too large or small, angles too far from straight ahead, or which indicate that the vehicle is completely off-road.

2.3. Edge tracking

Running RANSAC independently on each frame with the above modifications gives quite good results, but without enforcement of frame-to-frame consistency there are inevitably jumps between estimates. Moreover, the implication of RANSAC that no configuration other than one parametrized by three of the data points is possible makes the results highly dependent on the density of points along the road borders. Allowing estimated road segments that are only "close" to a sample is necessary for actual temporal continuity rather than simply temporal proximity.

Achieving continuity is a tracking problem, so we used a *particle filter* [12], which has a good ability to recover from mistracking such as might occur when ladar points become too sparse. We used a likelihood function $p(\mathbf{z}|\mathbf{x})$ that is proportional to the number of inliers according to the RANSAC criteria above and $n = 500$ particles. Properly, a road segment requires 4 parameters, but if we rule out roads orthogonal to the direction of travel, we can represent it with a 3-parameter state $\mathbf{x}$: angle θ, width w, and the lateral offset Δx from the vehicle's origin.

A shortcoming of this approach is that when ladar obstacles are sparse as in Figure 2(f), the correct road angle θ cannot be unambiguously determined. The tracker recovers nicely when sufficient ladar points return, but during such episodes the indicated road region is not reliable. We have previously used a vision-based road vanishing point detection method [13] that yields excellent θ estimates in nearly all situations, so the ladar road segment tracker just takes θ from this module and itself only estimates a 2-parameter state consisting of the road width and lateral offset.

We briefly review the method in [13]. To obtain the road vanishing point, the dominant texture orientation at each pixel of the low-resolution image is estimated by convolution with a bank of 12×12 complex Gabor wavelet filters [14] over 36 orientations in the range $[0, \pi]$. The filter orientation $\theta(\mathbf{p})$ at each pixel $\mathbf{p}$ which elicits the maximum response implies that the vanishing point lies along the ray

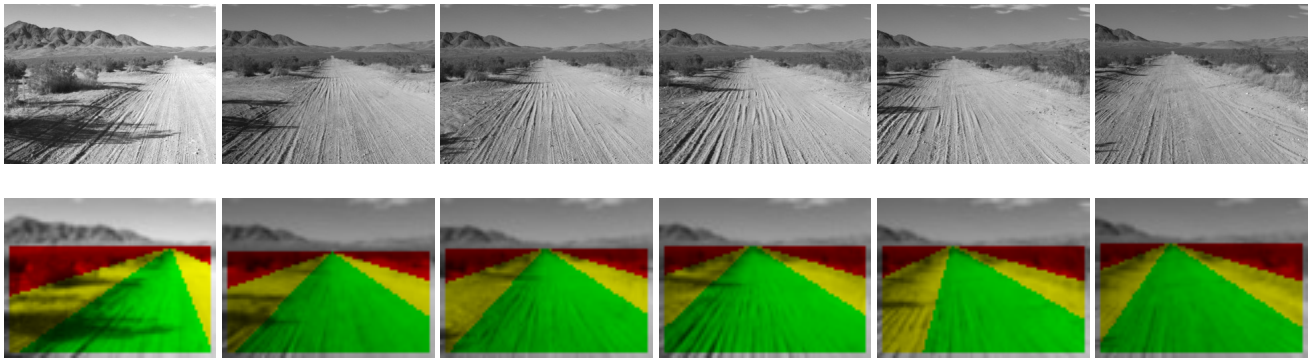


Fig. 3. Ladar- and vanishing-point-derived road regions for a ~ 2.5 second sequence at equal intervals as the vehicle swerves from left to right. Green pixels are in the road region, red off-road, and yellow are unlabeled pixels considered too close to the boundary.

defined by $\mathbf{r_p} = (\mathbf{p}, \theta(\mathbf{p}))$. The instantaneous road vanishing point for all pixels is the maximum vote-getter in a Hough-like voting procedure in which each $\mathbf{r_p}$ is rasterized in a roughly image-sized accumulation buffer $\mathbf{I}_{VP}$. Using $\mathbf{I}_{VP}$ as a likelihood function, we track the road vanishing point over time with a particle filter.

2.4. Label assignment

Projecting the tracked left and right ladar edge lines into the image immediately indicates a triangular road region as the blue lines in Figure 2(b) and (e) show. To get the upper limit of the left and right non-road regions, we use the horizon line (the magenta lines in the same figures) indicated by our vanishing point tracker from [13]. We can assume that pixels above the horizon are part of the sky and not consider them further.

However, because of uncertainty in the ladar road segment tracker, the inherent error of performing linear fitting when the actual road edge may be curved, and some aspects of the calibrations that guide the ladar-to-image transformation, we assume that the labels of pixels near the borders between regions are highly uncertain. Thus, we designate two regions of *unlabeled* pixels defined as all points within some distance d in vehicle coordinates of the appropriate left- or right-side line (we use $d = 2$ meters here).

Examples of automatically-labeled (and unlabeled) pixels are shown in Figure 3 for a short sequence of images. Green pixels are thought to belong to the road class, red to the non-road class, and yellow are unlabeled.

2.5. Patch classification

Given the positive and negative labels from the road segment tracker described above, we can attempt to learn a model of the appearance characteristics of small neighborhoods in each region. This will lead to predictions for the unlabeled pixels, as well as possibly changing some of the labeled pixels' class memberships if they are inconsistent with the learned class characteristics.

For efficiency, a low-resolution (80×60) image was used to compute features. For this work we used a grayscale CCD, so color was not an option for features. We experimented with two sets of features which we will call `Texture` and `Raw`. The 39 `Texture` features at a pixel (x, y) consist of the intensity of the pixel, the mean intensities over 5×5 and 11×11 pixel neighborhoods around the pixel, and the responses of the 36 Gabor filters from the vanishing point tracker. The thought here is that texture anisotropy alone, plus a very basic sense of the lightness or darkness of the neighborhood, may be sufficient to discriminate road patches (light, rutted) from background (mostly darker, less-oriented vegetation). The 25 `Raw` features are simply the intensities of the pixels in the 5×5 patch surrounding each pixel. The justification for this is that average lightness and darkness can be computed, texture information is available if the classifier can exploit it, and any other small-scale features that help with discrimination can be "discovered."

We tried several basic classification algorithms, including support vector machines (SVM) [15] and multilayer perceptrons (MLP) [16]. The SVMlight package [17] was used for SVM training; the kernel used for all of results here was linear. Matlab's Neural Network Toolbox was used to create the MLPs. Standard sigmoidal gate analog neurons were used, and backpropagation employed for training. Automatic input scaling was done by Matlab. One hidden layer with five neurons fed into a single output neuron.

These algorithms were chosen not for purposes of comparing them but rather to demonstrate that the ladar-derived labels we generate are good enough for off-the-shelf, untuned classifiers to get good performance from them.

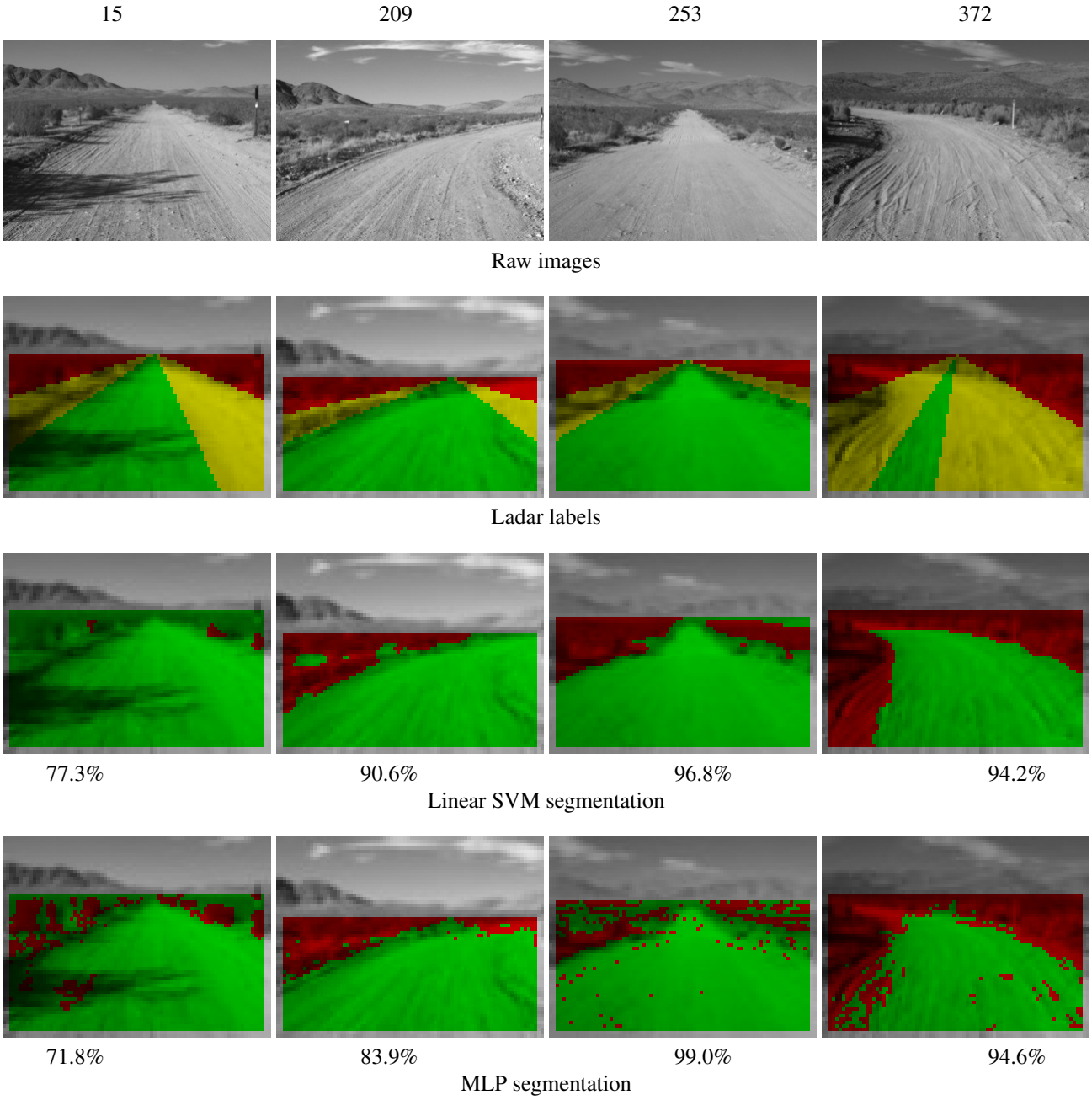
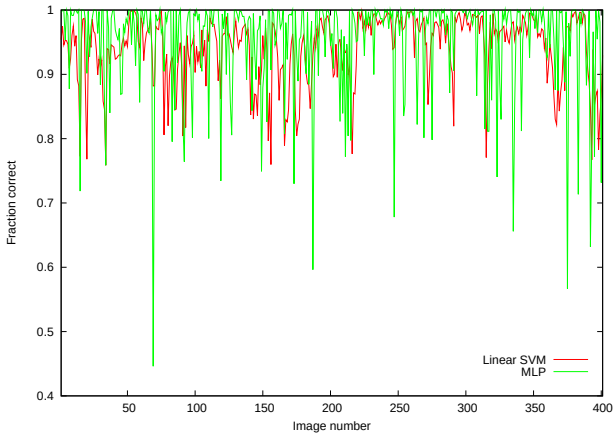


Fig. 5. Selected images, their automatic labelings, and final segmentations. Image numbers head each column, and the numbers under the segmentation images give the accuracy rate when that classifier is tested on the training feature vectors for that image. The segmentation images also show the assigned labels for the unlabeled pixels.

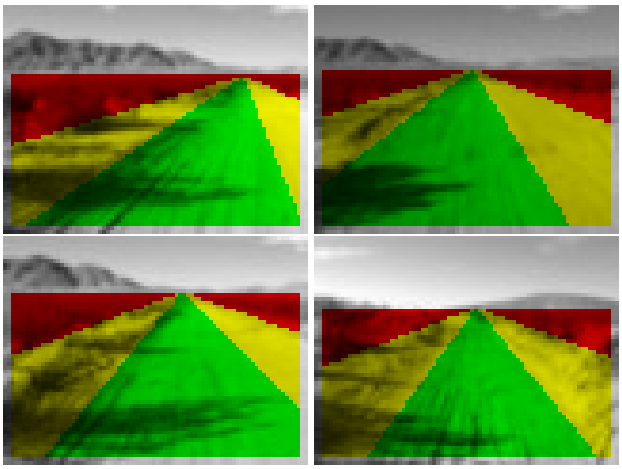
3. RESULTS

For a baseline comparison between feature types, we ran the system on 100 consecutive frames captured at 30 Hz. Training and testing on the same labeled feature vectors of each image independently, the SVM yielded a median error rate of 11.0% on `Texture` and 3.2% on `Raw`.

Based on this, we ran several larger tests just using `Raw` over a sequence lasting about 3.5 minutes with many turns, shadows across the road, and areas of sparse obstacles. 401 images separated by 0.5 second intervals were used; independent SVM and MLP classifiers were trained on the la-



(a)



(b)

Fig. 4. Image difficulty: (a) Accuracy of linear SVM vs. MLP classifiers when tested on same 401 training images (using `Raw` feature vectors); (b) Images producing four lowest SVM accuracies (clockwise from upper left: image 20, 34, 393, 156) all have strong shadows across road

beled features of each image. Testing again on the training images, linear SVM yielded a median error of 4.4% and MLP of 1.0%. The per-image results are plotted in Figure 4(a). Although the MLP had a lower median error, its worst error was 55.4% and it exhibited an error $\geq 25\%$ in 13 images. The SVM classifier's worst error was 24.1%. We did not see any instances of the labels provided by the ladar road segment tracker being misleading, and there seemed to be no visual reason for the MLP's lowest accuracies. However, the images that the SVM did worst on (four are shown in Figure 4(b); see also the first column of Figure 5) consistently had large shadows across the road. Shadows on the road with similar intensities to the off-road areas cause a bimodal distribution of intensities among road-labeled feature

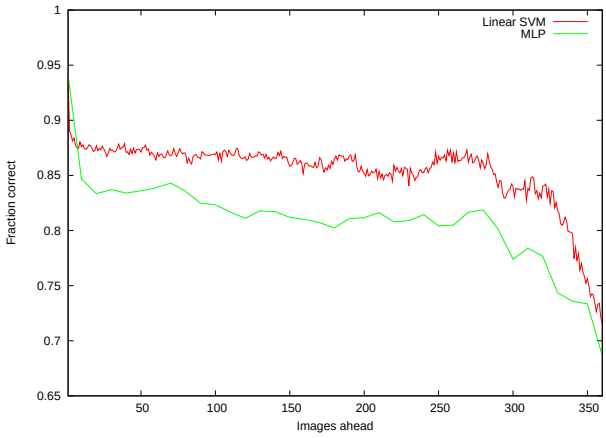


Fig. 6. Future validity of appearance models: Mean accuracy of classifiers as a function of time difference with test image (`Raw` feature vectors). MLP classifiers were only tested at 10-image intervals.

vectors that cannot easily be separated by the SVM's linear kernel. The MLP with its nonlinear decision boundary had less of a problem with shadows. Color features would certainly help both classifiers here.

Selected images and segmentations from this set are given in Figure 5. Images 209, 253, and 372 show that the very coarse geometry indicated by the ladar-derived labels can be improved upon by the pixel classifiers. In each case the road is curving–right, up, and left, respectively–in a way that the linear fitting of the ladar road segment tracker alone cannot capture. Postprocessing on these segmentations (e.g., recursive curve fitting a la [18]) would likely be able to explicitly infer the road curvature and aid the vehicle controller to plan steering and braking more precisely.

Another characteristic we are interested in is how valid the learned road appearance models are for other frames. In general, road appearance changes slowly, but phenomena like shadows or surface material boundaries can violate such assumptions. Ideally, the road segmenter would learn a new model from the ladar for each new image, but depending on the classifier and features used, doing this in real time may not be possible. Also, there may be extended periods in which roadside obstacles are sparse, rendering the ladar labels uninformative at best. In either case, older appearance models would have to be used. Our confidence in such models depends on how quickly their validity is expected to degrade as time passes.

Figure 6 shows some statistics related to this question. Here we took the classifiers trained on each of the 401 images as described above and tested them on images progressively later in the sequence. The horizontal axis shows the time difference Δt and the vertical axis shows the mean accuracy over all training images. Because the number of suc-

ceeding images is small toward the end of the sequence, the mean is noisy and thus we do not plot image differences beyond 360. As expected, accuracy declines as the time difference increases, but not very quickly. After an initial drop, the error rate of the linear SVM stays between 10% and 15% up to two minutes in the future. The MLP, measured at 10-image intervals, tracks the same general trend while performing slightly worse.

4. CONCLUSION

We have demonstrated that a visual road appearance model can be learned online from an automatic ladar-labeling procedure. An initially coarse segmentation with linear boundaries is refined to fit nonlinear shape features efficiently and without without manual intervention. Preliminary indications are promising and point toward an ability to deal with gradual changes in road material and illumination conditions.

Along these lines, we have begun testing performance when the classifiers have some "memory" by incorporating features from past frames. The train-on-one-frame method our results cover here seems to work relatively well for segmenting larger sets of frames, but training on multiple images would be expected to increase robustness in the face of abrupt changes. One approach is to learn a comprehensive model from numerous examples offline, and then for efficiency "track" a subset of the support vectors that are currently most applicable [9]. However, we have new labeled feature vectors coming in with each image frame with which to update the road appearance model. Conversely, we would like to forget older data that is not currently representative of the road. One simple approach is to use a sliding window of data from the present time t to $t-\Delta t$ and to completely relearn a model over that window with each tick of the clock. This approach is discussed for SVMs in [19] with particular attention to the size of the window used. A fixed-size window presumes that category boundaries change at a constant rate, which is not necessarily true. Another issue is that relearning *de novo* at each time step can be costly–is it possible to update the model more efficiently when new data arrives? This is examined for SVMs in [20] and related papers.

We have also experimented with a classifier based on *parallel perceptrons* (PP) [21], a single layer of simple perceptron neurons without lateral connections that can be used for complex classification tasks. This architecture, combined with an extension of the classic delta rule called the *parallel delta rule* (p-delta rule) has shown results comparable to MLPs trained with gradient descent algorithms such as backpropagation. PPs are also of interest to us because of their use as the training mechanism for systems based on Liquid State Machines (LSMs) [22], which may also help address some of the temporal aspects of the problem.

5. REFERENCES

[1] C. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.

[2] B. Southall and C. Taylor, "Stochastic road shape estimation," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 205–212.

[3] N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.

[4] J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2496–2501.

[5] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.

[6] J. Zhang and H. Nagel, "Texture-based segmentation of road images," in *Proc. IEEE Intelligent Vehicles Symposium*, 1994.

[7] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[8] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," in *Robotics: Science and Systems Conf.*, 2005.

[9] S. Avidan, "Subset selection for efficient SVM tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 85–92.

[10] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robotics: Science and Systems*, 2006.

[11] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29, pp. 5–28, 1998.

[13] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[14] T. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[15] N. Cristiani and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[16] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1996.

[17] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, 1999.

[18] E. Dickmanns and B. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 14, pp. 199–213, 1992.

[19] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," in *Inter. Conf. on Machine Learning*, 2000.

[20] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Conf. on Neural Info. Proc. Systems*, 2000.

[21] P. Auer, H. Burgsteiner, and W. Maass, "Universal learning with parallel perceptrons," Tech. Rep. NC-TR-01-111, NeuroCOLT, 2001.

[22] W. Maass, T. Natschlager, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.