CAMERA-BASED CLEAR PATH DETECTION

Qi Wu[1], *Wende Zhang*[2], *Tsuhan Chen*[3], *B.V.K. Vijaya Kumar*[1]

Electrical & Computer Engineering Department, Carnegie Mellon University[1]
Electrical Controls and Integration Lab, General Motors[2]
School of Electrical and Computer Engineering, Cornell University[3]

ABSTRACT

In using image analysis to assist a driver to avoid obstacles on the road, traditional approaches rely on various detectors designed to detect different types of objects. We propose a framework that is different from traditional approaches in that it focuses on finding a clear path ahead. We assume that the video camera is calibrated offline (with known intrinsic and extrinsic parameters) and vehicle information (vehicle speed and yaw angle) is known. We first generate perspective patches for feature extraction in the image. Then, after extracting and selecting features of each patch, we estimate an initial probability that the patch corresponds to clear path using a support vector machine (SVM) based probability estimator on the selected features. We finally perform probabilistic patch smoothing based on spatial and temporal constraints to improve the initial estimate, thereby enhancing detection performance. We show that the proposed framework performs well even in some challenging examples with shadows and illumination changes.

Index Terms— Computer vision, Feature extraction , Object Detection, Smoothing methods , Autonomous vehicles,

1. INTRODUCTION

The traditional methods for autonomous driving first detect all objects (e.g., vehicles, pedestrians, buildings, and trees) in the scene, and infer the remaining area as clear path with an assumption that the no-object area is the feasible region for autonomous driving. During the last decade, several object detection methods have been introduced in the literature. High-cost solutions using active sensors (such as Radar [1] and LIDAR [2]) show promising results for object detection in the autonomous vehicle competition, 2007 Defense Advanced Research Projects Agency (DARPA) Urban Challenge. Low-cost solutions using passive sensors (such as cameras), combined with computer vision algorithms, offer more affordable and no-interference solutions which also track objects reasonably well. [3] used stereo-vision-based methods to detect vehicles and objects. [4] achieved vehicle and pedestrian detection by learning information from the motion and edge cues. [5] modeled the statistics of object appearance and non-object appearance by two histograms of wavelet coefficient codewords. [6] adopted spatial-temporal filters based on shifted frame difference to augment the pedestrian detection using spatial filters alone, which considered both motion and appearance to improve detection performance.

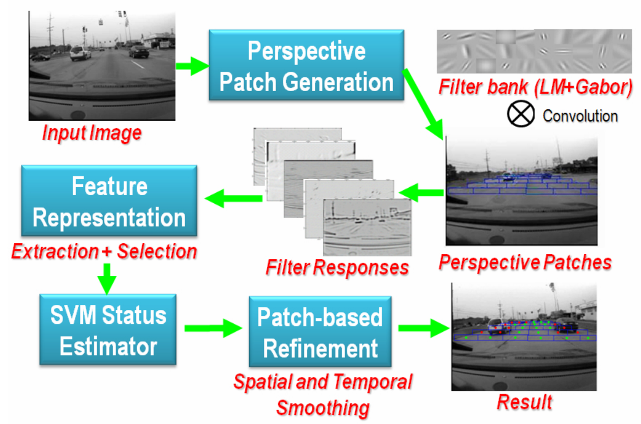


Fig. 1. System Overview

However, generic detection (i.e., detecting all kinds of objects) is a challenging task. The object appearance varies between different classes. Intra-class variation in each class also makes the detection less reliable. In addition, object appearance also varies depending on the host vehicle motion, lighting, and weather, which makes multiple-object detection systems complex.

In this paper, we turn the problem around. We detect the clear path, whose features are clustered together due to its similar texture, directly for autonomous driving. Therefore, we use only one clear path detector instead of a combination of multiple object detectors. We will show through examples that this approach has the potential to achieve improved clear path detection. Fig.1 gives an overview of the system, comprising of four components: perspective patch generation, feature extraction and selection, support vector machine (SVM) status estimator, and patch-based refinement. In addition, compared to the traditional detection algorithms, there are two novel aspects in the proposed method. 1) *Perspective Patch:* We generate rectangular patches on the ground in the world coordinates and project them to the image coordinates for computational efficiency. And 2) *Patch-based Smoothing with Spatial and Temporal Constraints:* The patch smoothing method enforces the spatial and temporal constraints of texture consistency.

The paper is organized as follows. In the next section, we introduce perspective patch generation. We discuss initial clear path estimation and patch-based smoothing in Section 3 and Section 4, respectively. In Section 5, experimental results are shown that the clear path detection approach delivers high accuracy, and conclusions are given in Section 6.

2. PERSPECTIVE PATCH GENERATION

In traditional object detection applications, there are two kinds of patches in images without considering any perspective information: fixed-grid patch [5] and dynamic-size patch [6], since objects are perpendicular to camera's optical axis. However, the clear path lies on the ground and is parallel to the camera's optical axis. Instead of defining patches in image coordinates, we define the patches in the world coordinates lying on the ground as shown in the Fig.2 and project them to the image coordinates considering the perspective of clear path.

In our proposed method, we first define the clear path candidate region in the world coordinates with a 9x25 (meters) rectangle in front of the vehicle (Fig.2(A)). Secondly, this region is divided into small patches. The sizes of these patches are not equal. The faraway patches are long and the near-by ones are short in the longitudinal direction (with an 8-meter length for the farthest and a 2-meter length for the closest), since the projected image patches have to be large enough for accurate classification. The camera calibration parameters and the pinhole camera model are used to project the ground patches to the image coordinates as quadrilaterals in Fig.2(B). We approximate quadrilateral patches as rectangular patches for computational efficiency as shown in Fig.2(C), which are the perspective patches.

3. INITIAL CLEAR PATH ESTIMATION

Initial clear path estimation contains two stages: feature representation and learning. We first extract discriminative texture features to distinguish clear path vs. obstacles. Each perspective patch is convolved with an extended Leung-Malik filter bank (78 filters mixed with edge, bar and spot filters at multiple orientations and scales) and Gabor filter bank (90 filters at 9 directions with different parameters). We sum all absolute responses of each filter within a patch as a texture value. After normalization, we represent each patch with a 168 dimensional feature vector. Then, we adopted Adaboost [7] to select the 50 most discriminative features for classification to improve the computational efficiency.

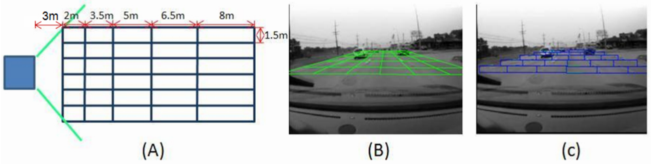


Fig. 2. Perspective Patch (A) Ground patches in world coordinates. (B) Projected ground patches in image. (C) Perspective patches.

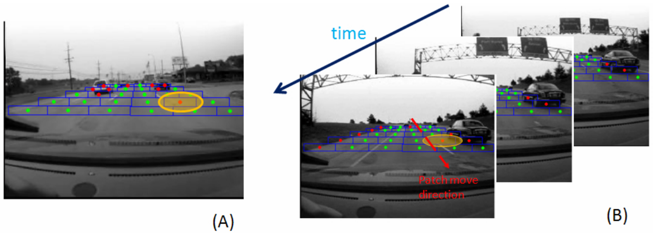


Fig. 3. Wrongly classified patches are circled. (A) Spatial Inconsistency. (B) Temporal Inconsistency.

In the learning stage, we first train the initial clear path estimator using Support Vector Machines (SVM) probability estimation [8] based on the perspective patch features. Then, in the test stage, this estimator provides the probability $P_j^0(c)$ of both classes ("clear path" and "obstacles") of each patch based on patch's features. Finally, we use maximal likelihood estimate $\langle \hat{c}_j^0 \rangle = \arg\max_c P_j^0(c)$ to identify each patch's initial SVM classified label of "clear path" ($\hat{c}_j^0 = 0$) or "obstacles" ($\hat{c}_j^0 = 1$). The initial probabilities and classified labels are used in the next section.

4. PATCH-BASED REFINEMENT

Each patch can be simply classified into 2 classes: "clear path" or "obstacles" by SVM. Sometimes, SVM makes wrong decisions due to the texture ambiguities of local perspective patches. There are two types of errors as shown in Fig.3:

1) *Type I:* Within the same frame, the clear path patch is wrongly classified as "obstacles", while it is surrounded by clear path patches with similar texture.

2) *Type II:* Between successive frames, the patch in the current frame is classified as "obstacles", while its corresponding vehicle-motion-compensated regions in the previous frames are all classified into "clear path" with similar texture.

Both errors can be corrected if we consider the spatial and temporal consistency between patches. Therefore, we refine patch initial probability $P_j^0(c)$ between its neighboring patches and between its corresponding regions at the previous frames iteratively. The probability of patch s_j to be clear path or not, $P_j^t(c)$, is updated iteratively as follows:

$$P_j^t(c) = \frac{n_j^t(c) \prod_{k \in N} c_{j,k}^t(c)}{\sum_{c \in \{0,1\}} n_j^t(c) \prod_{k \in N} c_{j,k}^t(c)}, \qquad (1)$$

where $n_j^t(c)$ is the spatial smoothing coefficient, which constrains neighboring patches with similar texture to be the same class. And $c_{j,k}^t(c)$ is the temporal smoothing coefficient which enforces patch's consistency in the projected regions at the previous k frames. The maximal likelihood estimate $\hat{c}_j^t$ of patch s_j at iteration t is: $\langle \hat{c}_j^t \rangle = \arg\max_c P_j^t(c)$.

4.1. Spatial patch smoothing

Let s_l denote one of the current patch s_j's neighboring patches with its associated initial probability $P_l^0(c)$ and max-

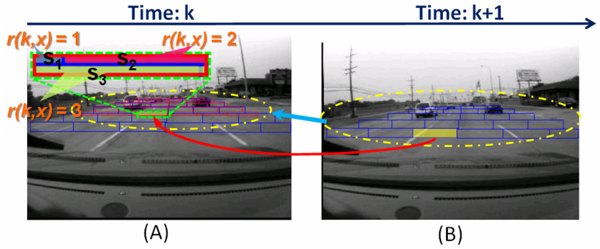


Fig. 4. Spatial Patch Smoothing. The patches in the current frame are projected onto the previous frame given the vehicle's speed and yaw rate.

imal likelihood estimate $\hat{c}_l^0$ obtained from SVM. The spatial smoothness enforces the constraint that neighboring patches with similar texture should have the same class c. Therefore, we model class similarity of the neighboring patches of patch s_j by a contaminated Gaussian distribution with mean $\hat{c}_l^{t-1}$ and variance σ_l^2 . We define spatial smoothness coefficient $n_j^t(c)$ to be:

$$n_j^t(c) = \prod_{s_l} N(c; \hat{c}_l^{t-1}, \sigma_l^2) + \varepsilon, \qquad (2)$$

where ε is a small constant (e.g., 10^{-10}) in case of division by zero. We calculate the variance σ_l^2 using 1) the texture similarity $N(\Delta_{j,l}; 0, \sigma_\Delta^2)$ of the patches $\Delta_{j,l}$, which measures the texture difference between patches s_j and s_l by Gaussian model, 2) the neighboring connectivity $b_{j,l}$, which contains the percentage of patch s_j's border between patches s_j and s_l, and 3) the probability of patch s_l: $P_l^{t-1}(c)$ obtained from the last iteration $t-1$. Hence, σ_l^2 is defined as $\sigma_l^2 = g/P_l^{t-1}(c)^2 b_{j,l} N(\Delta_{j,l}; 0, \sigma_\Delta^2)$, where g and σ_Δ^2 are constants ($g = 8$ and $\sigma_\Delta^2 = 20$ in our experiment). Therefore, if patch s_j and its neighboring patch s_l have similar textures, and patch s_j's class is consistent with its neighbor's label estimates (they are both classified as obstacles or clear path), we expect spatial smoothness coefficient $n_j^t(c)$ of patch s_j to be large.

4.2. Temporal patch smoothing

Given the vehicle's speed and yaw rate, we can map the locations of clear path patches in the current frame (blue rectangles in Fig.4(B)) to the previous k frames (red rectangles in Fig.4(A), $k = 1$) by assuming that they are stationary on the ground without any occlusion. The temporal smoothing coefficient $c_{j,k}^t(c)$ is used to ensures that the patch s_j's estimate is consistent with its corresponding estimates at the previous frames. If the obstacles occlude a clear path patch, we can not find the corresponding area in the previous frames. Therefore, in the proposed method, we define the temporal smoothing coefficient $c_{j,k}^t(c)$ of clear path based on temporal consistency, visibility, and patch s_j's initial probability obtained from SVM.

Temporal Consistency: Given the host vehicle speed v, yaw rate γ and frame rate of video f, we first calculate the host vehicle motion (distance v/f and yaw angle γ/f) between two neighboring frames. Then, we project the motion-compensated region to its neighboring frames using a pinhole camera model. Finally, since the projected region may cover several patches in previous frames, we calculate patch s_j's projecting distribution $b_{j,k}^t(c)$ based on the probability distribution at the projected previous k frames to estimate the temporal consistency without occlusion: $b_{j,k}^t(c) = 1/num_{s_j} \sum_{x \in S_j} P_{r(k,x)}^{t-1}(c)$.

As shown in Fig.4, the yellow patch at frame $k+1$ is mapped to the a smaller yellow patch at frame k via vehicle motion compensation. $r(k,x)$ is the patch index representing which patch the pixel x belongs to at frame k. And num_{s_j} is the number of the pixels on patch s_j. If the projected region's status is consistent with patch s_j's status, we expect $b_{j,k}^t(c)$ to be large when patch s_j is visible at the previous frame.

Visibility: Due to possible occlusions, a patch might not have the corresponding pixels in the previous frame. We estimate the likelihood $v_{j,k}$ to measure a patch's visibility. This likelihood is modeled by a Gaussian distribution represented by the texture similarity between the current patch and each projected region at the previous frame as follows: $v_{j,k} = N(\Delta_{j,k}; 0, \sigma_\Delta^2)$, where $\Delta_{j,k}$ is texture similarity of patches s_j and s_k, and σ_Δ^2 is a constant ($\sigma_\Delta^2 = 20$).

Now, we combine the visible and occluded cases. If the patch is visible, $c_{j,k}^t(c)$ is calculated from the visible consistency likelihood $b_{j,k}^t(c) P_j^{t-1}(c)$. Otherwise, its occluded consistency likelihood is a fixed prior P^t(e.g., $1/2$). Therefore,

$$c_{j,k}^t(c) = v_{j,k} b_{j,k}^t(c) P_j^{t-1}(c) + (1 - v_{j,k}) P^0. \qquad (3)$$

5. EXPERIMENTS

The videos for experiments were captured on a Cadillac CTS 2006 vehicle. The frame rate is 5 fps and the videos have 1478 frames at 320*240 pixels manually labeled clear path patches and non-clear path patches: 981 frames are used for training, and 497 frames are for test. In these videos, various conditions are covered. Some sample images are shown in Fig.5 in urban, highway, shadows and illumination change conditions. Our experiments confirm that we achieved reasonable performance in these situations.

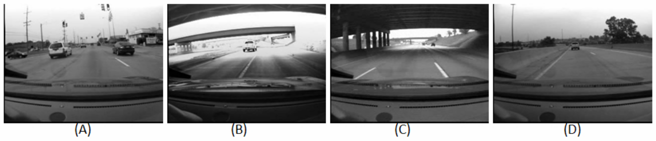


Fig. 5. Sample images from the database. (A) Urban (B) Illumination change (C)Shadow (D) Highway.

In the test stage, each frame is represented by 30 perspective patches with 50 features for each patch. The SVM classifier with RBF kernel and parameters $C = 32$ and $\gamma = 0.0313$ provides the initial probability estimation. Considering the computational efficiency, we only calculated the influence from the previous frame ($k = 1$). Fig.6 shows some detection results without smoothing in the urban and highway cases. The results show that the proposed algorithm distinguished clear path from different types of obstacles (e.g., vehicles

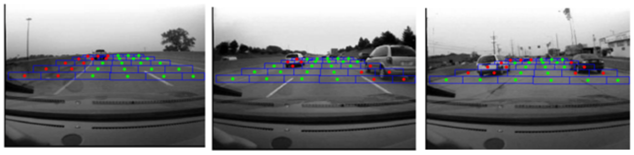


Fig. 6. Result of initial detection using SVM

or road-side) without any smoothing. However, SVM-based approach wrongly classified lane-markers as non-clear path as shown in Fig.7(A). Furthermore, in the challenging cases as shown in Fig.7(A,2) and Fig.7(A,3), the bridge's shadow and illumination change influenced a patch's texture causing a wrong decision. We applied spatial and temporal smoothing to further improve the detection performance. Fig.7(B) demonstrates the results after smoothing. For example, the wrongly classified patch in Fig.7(A,1) was corrected by the spatial smoothing of its neighboring patches. The wrongly classified patches in Fig.7(A.2) and (A.3) were corrected by the temporal and spatial smoothing of neighboring patches.

Fig.8 summarizes the performance of clear path detection with and without smoothing using ROC curve. Compared with SVM classification which had 92.23% in accuracy (the number of correctly classified patches / the number of total patches), 4.6% in FAR (False Alarm Rate) and 5.1% in FRR (False Rejection Rate), additional patch-based smoothing improved the accuracy to 94.57% and reduced FAR to 3.2% and FRR to 4.5%. Additional results are shown in Fig.9 which demonstrates our method performing well in various scenarios such as urban, countryside and highway.

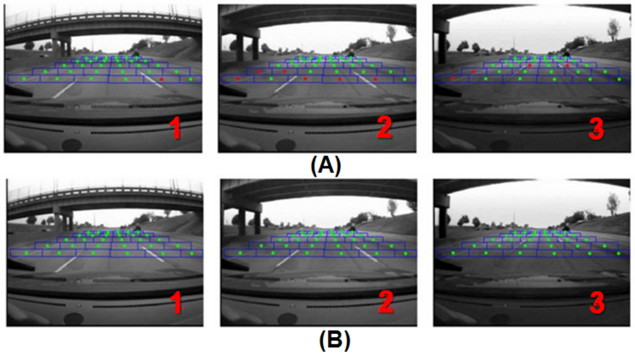
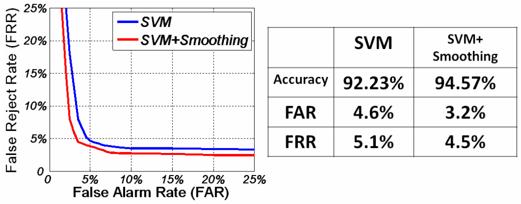


Fig. 7. Patch-based smoothing



	SVM	SVM+ Smoothing
Accuracy	92.23%	94.57%
FAR	4.6%	3.2%
FRR	5.1%	4.5%

Fig. 8. Comparison of clear path detection

6. CONCLUSIONS

Traditional methods for autonomous driving detect all objects in the scene, and infer the remaining areas as clear path. However, such a system, which requires multiple object detectors, is complex, slow and not very reliable.

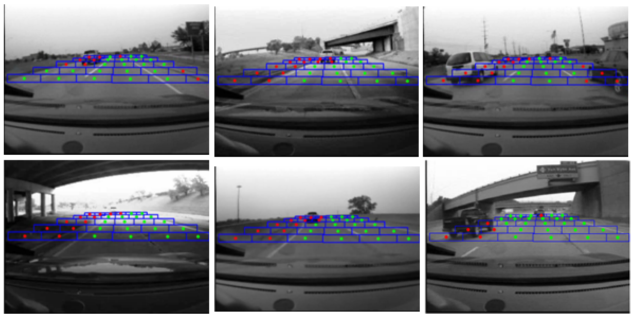


Fig. 9. Additional results in various scenarios

In this paper, we proposed a method to detect clear path directly in the scene only using one clear path detector and showed that it performed robustly even in some challenging situations with shadows and illumination changes via spatial and temporal smoothing.

7. REFERENCES

[1] S. Sugimoto, H. Takahashi, and M. Okutomi, "Obstacle detection using millimeter-wave radar and its visualization on image sequence," *IEEE international Conference on Pattern Recognition*, pp. 342–345, 2004.

[2] C. Wang, C. Thorpe, and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," *IEEE Intelligent Vehicles Symposium*, 2003.

[3] M. Salinas, E. Rafael, and F. Aguirre, "People detection and tracking using stereo vision and color," *Image Vision Computing*, pp. 995–1007, 2007.

[4] I. Alonso, D. Llorca, and M. Garrido, "Combination of feature extraction methods for svm pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, pp. 292–307, 2007.

[5] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 65–81, 2007.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 551, 2001.

[7] D. Karuppiah P. Silapachote and A. Hanson, "Feature selection using adaboost for face expression recognition," *International Conference on Visualization, Imaging, and Image Processing*, p. 551, 2004.

[8] T. F. Wu, C. J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, pp. 975–1005, 2004.