PARKING SPACE DETECTION FROM VIDEO BY AUGMENTING TRAINING DATASET

Wei Yu[1], *Tsuhan Chen*[2] *

[1]Carnegie Mellon University, [2] School of Electrical Computer and Engineering, Cornell University

ABSTRACT

Auto parking techniques are attracting more attention these days. In this paper, we develop an image-based method to estimate the depth contour in parking areas. Our algorithm is an extension of the canonical appearance-based models for object recognition. One challenge in object recognition is that limited training dataset can hardly represent all kinds intra-class and inter-class variations. We propose to augment the limited training dataset by on-the-spot learning from test data. The information is obtained by applying a fast block based stereo algorithm to estimate a rough disparity map. New "soft" samples are created to augment the training sample library. We present improved classification performance by using the proposed technique.

Index Terms— auto parking, object recognition, classification, stereo

1. INTRODUCTION

Automatic parking system is one of the interesting topics in developing intelligent vehicles. J.D. Power'2001 Emerging Technology Study shows that over 66% of consumers are likely to purchase parking assistance [1]. In this paper, we are interested in automatic detection of vacant parking space by analyzing the content of videos captured during driving process. The application scenario is illustrated in Fig 1. A camera is set up on the host vehicle, facing the sidewalk with axis $30°$ below the horizon and $50°$ vertical field of view (FOV). The host vehicle is moving at almost constant speed (less than 15mph) in the parking area. Our goal is to reconstruct the complete depth contour of the scene.

Some previous work [2] explores image analysis techniques for parking space detection using images taken from a fixed camera viewing from the top of a parking area. This would be helpful when such camera monitoring system is available. However, many parking areas don't have such systems.

The application is basically an image understanding task: to recognize object categories from images. In this application, object categories of interest are "vehicle" and "background". The problem of object recognition has been extensively studied. Two main issues in developing object

*Thanks to ITRI of Taiwan for funding.

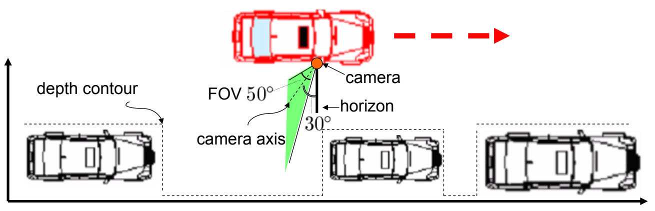

Fig. 1. Camera set up. The goal is to create depth contour.

recognition system are feature representation and classification scheme. We build our algorithm upon commonly used appearance-based model. More specifically, we follow the texton-based model developed in the context of texture recognition[3].

The fundamental challenge in object recognition is to handle a wide variety of transformations due to illumination change, viewpoint change, and intrinsic visual differences from the same object category. It's very difficult to find a compact representation of object categories which is invariant to intra-class variations and yet discriminative enough to distinguish inter-class differences. Also it's hard to learn all kinds of intra-class and inter-class variations from limited training dataset.

Another seemingly attractive approach is to reconstruct 3D geometry of the scene, which provides orthogonal information in addition to the appearance for object recognition. However, most state-of-art 3D reconstruction algorithms count on associating corresponding points in different views to retrieve depth, thus assuming Lambertian surface. This is not true for our application because car body and window exhibit strong specular reflection (mirror-like reflection).

Therefore we do not target at precise 3D reconstruction. Instead, we propose to enhance the training dataset by learning rough depth information on-the-spot from test data, thus specifically tune the appearance-based model to better fit the test data.

2. ALGORITHM DESCRIPTION

We detailed the proposed algorithm in this section. Section 2.1 describes the complete system flow on a high level. Section 2.2 briefly summarizes the texton-based model for rec-

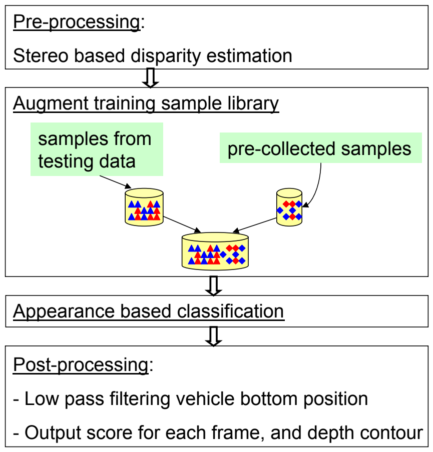
Pre-processing:
Stereo based disparity estimation
Augment training sample library
samples from testing data
pre-collected samples
Appearance based classification
Post-processing:
- Low pass filtering vehicle bottom position
- Output score for each frame, and depth contour

Fig. 2. Algorithm flow diagram

ognizing object class. Section 2.3 explains our proposal in details.

2.1. Algorithm Overview

The system consists of four major steps, as shown in Fig 2. First, in pre-processing step, a fast block matching based stereo method is applied to pairs of consecutive frames, to generate a rough disparity estimation, which is then used to create "soft" samples. Those "soft" samples are added to the pre-collected training dataset to augment the training sample library, which is used in appearance-based classification. In the end, post-processing step generates a score for each frame indicating how likely the frame is in class 1 ("vehicle") or 0 ("background"). Finally, a complete depth contour of the scene is created.

2.2. Texton-based Model

In texton-based model for object recognition, images are represented by histograms over a dictionary of "visual words" (also called "textons"). The visual words are K-means clustering centers of features. Features are filter responses generated by convolving the image with a filter bank. The filter bank is a set of orientation and spatial frequency selective linear filters. We follow [4] to use the one made of 3 Gaussians ($\sigma = 1, 2, 4$), 4 Laplacian of Gaussians (LOG) ($\sigma = 1, 2, 4, 8$), and 4 first order derivative of Gaussians (DOG) ($\sigma = 2, 4$ for both $x-$ and $y-$ directions). All filters are 5x5 windows. All filters in the filter bank are applied to a gray image to create $3 + 4 + 4 = 11$ dimensional feature vectors.

Features are computed on every pixel, aggregated over all training images and then clustered using K-means approach. The cluster centers (textons) define a visual dictionary. We use the dictionary size of 50 textons. After filtering an image, each pixel is mapped to the closest texton in the dictionary, thus a normalized histogram of textons can be generated on a region or image basis.

We apply the texton-based model on a superpixel basis. Every frame is segmented into superpixels using graph based segmentation [5]. Each superpixel is a data sample represented by a histogram of textons. In training stage, all manually labeled data samples (pre-collected samples) are collected in a sample library .

2.3. Proposed Method

To augment the training sample library, we generate new "soft" samples from test data.

Pre-Processing Stereo processing the test video is the first step. Every frame is paired with its next frame. We use block based matching stereo because it's fast and good enough for a rough estimation.

Every frame is resized to 300×200, and the block size used is 8×8. For all 8×8 blocks whose center (x, y) are multiples of 4, we search in the same row for the best matching block in the next frame, with disparity value ranging from 0 to 31 pixels. Only when sum of square difference (SSD) of the best matching block is obviously better than the second best match, that is, ratio of the best matching SSD to the second best is less than 0.8, then centers of two matching blocks are counted as a pair of correspondences with valid disparity estimation. Fig 3 shows an example disparity map (brighter pixels indicate larger disparity, black pixels indicate no disparity estimate).

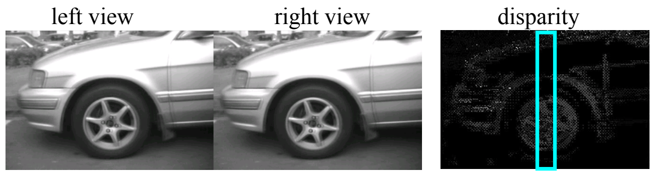

Fig. 3. block based stereo matching and disparity result.

There are two ways to distinguish two classes: 1)"vehicle" is likely to have larger disparity than average, 2) in "background" image, disparity of the upper region of the image is smaller than that of the lower region. We implement the idea as following (Fig 4).

1. Average disparity (Fig 4(a) cyan curve) of each frame is computed as the mean of all valid disparities in the middle of the image. The middle is defined as in the cyan box in Fig 3, with 8 pixels extension to the left and right from the center of the image. D1 (Fig 4(b) cyan curve) is low pass filtered average disparity. Average disparity in the upper 2/3 and lower 2/3 region of the image is computed in the same way, illustrated as D2 and D3 in Fig 4(b).

2. If a frame i satisfy $D1(i)/average(D1) > 1.4$ or $D2(i)/D3(i) > 1$, then the frame is assigned label 1 ("vehi-

cle"); otherwise the frame is assigned label 0 ("background"). The class label is shown in Fig 4(c).

3. Low-pass filter the class label to give a score for each class (Fig 4(d)), higher score indicates the frame more likely to contain "vehicle" in the middle cyan box.

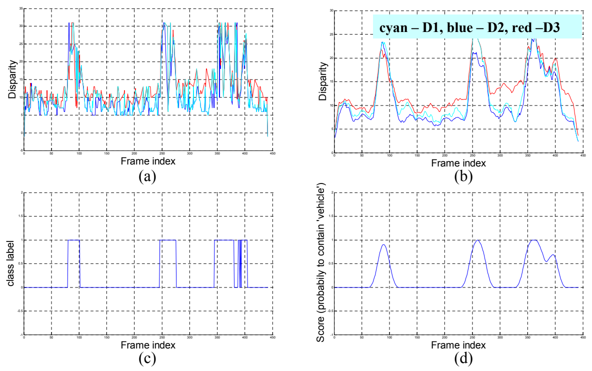

Fig. 4. (a) average disparity (cyan: full image, blue: upper 2/3 image, red: lower 2/3 image;). (b) lower pass filter (a). (c) assigned class label (1:"vehicle", 0:"background"). (d) low pass filter (c).

Augment training dataset In pre-processing step, a score is assigned to each frame, indicating how likely it contains "vehicle" in the middle. After segmenting an image into superpixels, if a superpixel's center falls into the middle cyan box or more than half area of the superpixel falls into the cyan box, the superpixel is added to the training sample library, associated with the frame score indicating its probability of coming from class 1. These new "soft" samples originate from the test data, thus able to captures the characteristics of the test data. The "soft" samples are combined with pre-collected samples to create a larger training sample library.

Classification We do experiments with two commonly used classification schemes: nearest neighbor and Gaussian model . When using nearest neighbor classifier, each superpixel finds its closest 10 samples in the training sample library. The average score of the 10 samples is the score for the test sample. If the score greater than 0.5, the test sample is assigned label 1; otherwise assigned label 0.

When using Gaussian model, we model the set of histograms for each class using Gaussian distribution by

$$P(\mathbf{H}|C) = \prod_{i=1}^{T} \mathcal{N}(H_i|\overline{H_{Ci}}, \beta_{Ci}^{-1}) \tag{1}$$

where H_i denotes the i^{th} bin of a histogram $\mathbf{H}$, T denotes the size of the visual dictionary, $\overline{H_{Ci}}$ and β_{Ci} denote mean and variance of distribution of the i^{th} bin in class C.

With the augmented training sample library, $\overline{H_{Ci}}$ and β_{Ci} are learnt as following:

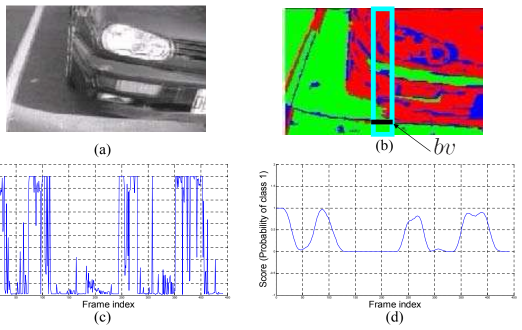

Fig. 5. (a) test frame (b) classification output (c) bv curve (d) final score for each frame

$$\overline{H_{1i}} = \frac{\sum_{\mathbf{H}} w_{\mathbf{H}} \cdot H_i}{\sum_{\mathbf{H}} w_{\mathbf{H}}}, \overline{\beta_{1i}} = \frac{\sum_{\mathbf{H}} w_{\mathbf{H}} \cdot (H_i - \overline{H_{1i}})^2}{\sum_{\mathbf{H}} w_{\mathbf{H}}} \tag{2}$$

$$\overline{H_{0i}} = \frac{\sum_{\mathbf{H}} (1 - w_{\mathbf{H}}) \cdot H_i}{\sum_{\mathbf{H}} (1 - w_{\mathbf{H}})}, \overline{\beta_{0i}} = \frac{\sum_{\mathbf{H}} (1 - w_{\mathbf{H}}) \cdot (H_i - \overline{H_{0i}})^2}{\sum_{\mathbf{H}} (1 - w_{\mathbf{H}})} \tag{3}$$

$w_{\mathbf{H}}$ is the score of sample $\mathbf{H}$, which is the probablity of this sample in class 1. In classification step, $P(\mathbf{H}|C)$ is computed using Eq (1), and the superpixel is assigned label $\hat{C} = \text{argmax}_C\, P(\mathbf{H}|C)$.

It would be difficult to evaluate results by checking the classification result on a superpixel basis, because that requires to provide ground truth class label for every superpixel in every test frame. Each video includes about 1000 frames and each frame includes about hundreds of superpixels, making manual labeling the ground truth a hard task. Instead, we do the following post-processing to output a score for each frame, and evaluates results on a frame basis.

Post-processing If a frame contains a vehicle in the center, the bottom position of a vehicle is usually large ($1/3$ of image height); otherwise most region is ground and the bottom position of a vehicle is very small (often 0 or slighter larger than 0). The bottom position bv is estimated using the maximum row index of pixels in class 1 in the middle cyan box, as shown in Fig 5(b). Connecting bv for all frames generates the plot in Fig 5(c). If $bv(i) > im\_height/3$, frame i is labeled 1; otherwise labeled 0. Then the class label is low pass filtered to generate a score for each frame as shown in Fig 5(d). This score is used for ROC analysis in section 3.

To create a depth contour, a "hard" label is assigned for each frame. Frames with a score higher than 0.5 is labeled 1, and 0 otherwise. Average $(im\_height - bv)$ of a segment of continuous frames assigned label 1 approximates distance between the "vehicle" in those frames and the host vehicle. Average $(im\_height - bv)$ of all frames assigned label 0 approximates distance of the "background" from the host vehicle. Finally, the distance of all frames are connected to create a "depth contour" of the complete scene as shown in Fig 6, providing driver with an “overview” of the parking area.

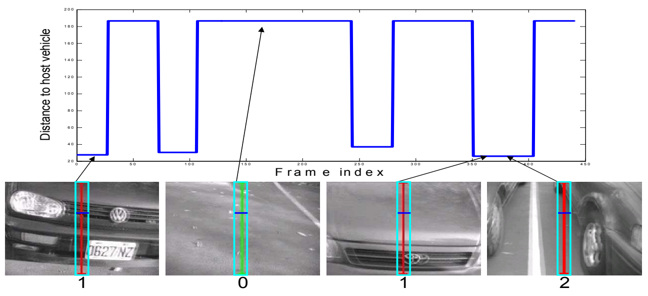

Fig. 6. Reconstructed depth contour. Center bar shows label assigned in post-procesising(red for class 1, green for class 0). Ground truth labels are shown below the frame.

3. EXPERIMENTS

Training dataset are 80 frames randomly sampled from two reverse parking videos. We manually label the “vehicle” region in each frame. There are totally 1368 samples in class 1 and 1165 samples in class 0 in the training dataset. Testing dataset includes 4 parallel parking videos and 3 reverse parking videos, captured in different parking areas. Average video length is about 1000 frames for both training and testing.

The ground truth of each test frame is manually labeled. We check the middle cyan box of each frame, if all pixels are “background”, it’s labeled class 0; if all pixels in upper 1/3 of the cyan box are “vehicle”, it’s labeled 1; otherwise, it’s labeled class 2 meaning ambiguous class. Some examples are shown in Fig 6. More than 99.5% of the frames are not labeled class 2. In the following, we only evaluate quantitative result on those “un-ambiguous” frames.

We do four comparison experiments, with two choices of the pre-collected training dataset and two choices of the classification scheme. For the pre-collected training dataset, either the complete training dataset is used or only partial (randomly selected 1/32) of it is used. For the classifier, either nearest neighbor (NN) or Gaussian model (GM) is applied. In each experiment, we compare the performance of two cases: with and without augmenting the training sample library. The classification and post-processing steps are exactly the same for both cases. Fig 7 presents ROC curve characterizing TPF (True positive rate) versus FPF (False positive rate). Table 1 shows quantitative results of ROC area and equal error rate. In all experiments, augmenting the training sample library shows better performance. It’s worth noticing that GM performs much better NN without augmentation when training dataset shrinks. This is because GM is a parametric model, while NN relies on large number data points to capture class variations.

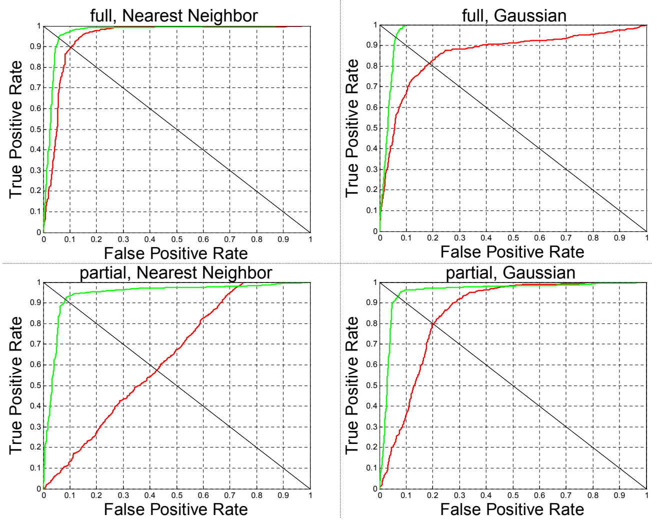

Fig. 7. ROC curve when using full and partial (1/32) of the training dataset. Each case tested with (*green*) and w/o (*red*) augmenting the training dataset.

Table 1. ROC area and Equal Error Rate

		full		partial	
		ROC area	EER	ROC area	EER
NN	w/o aug.	0.94	0.11	0.63	0.48
	with aug.	0.97	0.06	0.94	0.08
	boost	0.03	0.05	0.31	0.40
GM	w/o aug.	0.86	0.19	0.85	0.20
	with aug.	0.97	0.06	0.95	0.07
	boost	0.11	0.13	0.10	0.13

4. CONCLUSION

In this paper, we propose to augment the appearance-based classification model by on-the-spot learning from the test data. We apply this idea in the application scenario of detecting parking space. Experiments show improved recognition performance by using the proposed technique.

5. REFERENCES

[1] Randy Frank, “Sensing in the ultimately safe vehicle,” in *Convergence Conference and Exhibition*, 2004.

[2] Q. Wu C. Huang and T. Chen, “Robust parking space detection considering inter-space correlation,” in *ICME*, 2007.

[3] T. Leung and J. Malik, “Representing and recognizing the visual appearance of materials using three-dimensional textons,” in *IJCV*, 2001.

[4] J. Winn A. Criminisi and T. Minka, “Object categorization by learned universal visual dictionary,” in *ICCV*, 2005.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, “Efficient graph-based image segmentation,” in *IJCV*, 2004.