PRIOR-BASED VANISHING POINT ESTIMATION THROUGH GLOBAL PERSPECTIVE STRUCTURE MATCHING

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 this paper, we describe a prior-based vanishing point estimation method through global perspective structure matching (GPSM). In contrast to the traditional approaches which require an undistorted image with straight roads for vanishing point estimation, our method first infers vanishing point candidates of an input image from an image database with pre-labeled vanishing points. An image-based retrieval method is used to identify the best candidate images in the database by matching image's perspective structure. The initial estimation of input image's vanishing point is calculated from the pre-labeled vanishing points of the best candidates. Probabilistic refinement (PR) is then used to optimize the vanishing point estimate. Experimental results show that the proposed method works well in a variety of on-road driving environments (e.g., in urban, highway and country-side areas), especially with traffic captured by a fish-eye back-aid camera.

Index Terms— Image representation, Image matching, Computer vision, Autonomous vehicles

1. INTRODUCTION

The problem of vanishing point estimation has been addressed many times in the past decade. Most of the existing approaches fall into two categories: edge-based methods and texture-based methods. Edge-based methods, e.g., in [1], identify line segments (lane markers and road boundaries) in edge detector outputs and determine the vanishing point by searching a point close to most line segments. They work well on engineered roads such as highways that are painted with parallel lines clearly, but fail in scenes that possess neither strong edges nor contrasting local characteristics. Texture-based methods, e.g., in [2, 3], analyze the dominant orientation of texture in the image to identify dominant orientation rays and determine their intersection as the vanishing point. However, these approaches have difficulties in handling complex scenes (e.g. urban scenarios), where image features (e.g. lines or rays) might not all be on parallel lines that help to deduce the correct vanishing point. Moreover, both approaches are sensitive to the noise. We also need a better solution when applying these vanishing point methods to real road scenes with traffic or the scenes captured by a camera with unknown lens distortion.

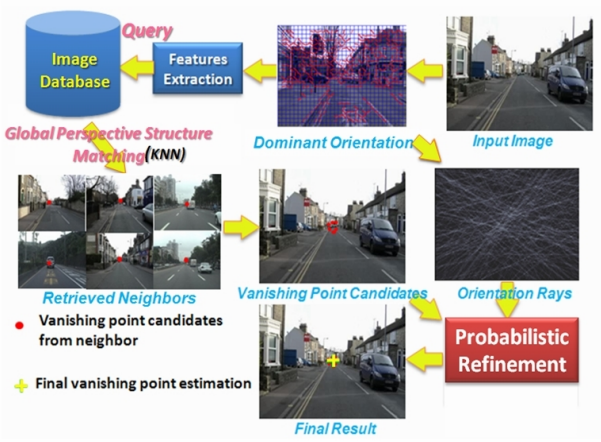

Fig. 1: Proposed Vanishing Point Estimation Approach

In autonomous driving application, we may access to a large image database of various road scenes with manually labeled vanishing points and it is reasonable to estimate an image's vanishing point position while taking advantage of the prior knowledge embedded in the database. Gallagher [4] proposed a method to use the pre-labeled vanishing points in the database to help determine vanishing point positions in the test image. He first identified the line segments in an image through edge analysis and then identified their intersections. Based on similarity measurement, each intersection was assigned a probability of being coincident with a pre-labeled vanishing point in the image database. Finally, weighted clustering is then performed to determine the most likely vanishing point which could lead to more precise estimation. However, his method is sensitive to noisy lines and lacks the ability to deal with camera lens distortion.

Since similar global perspective structure in different images may imply similar vanishing point positions, instead of computing numerous intersections as the vanishing point candidates [4], we propose an approach to identify fewer best vanishing point candidates directly from the neighboring images in the database as prior knowledge with the closest global perspective structure. An overview of this system

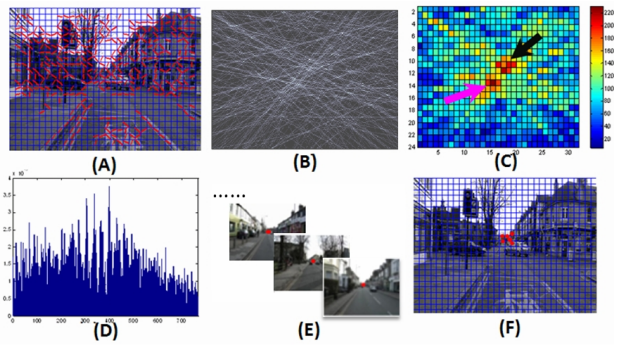

Fig. 2: Global Perspective Structure Matching (GPSM) for vanishing point initialization. (A) Dominant orientations on strong texture patches (B) Orientation rays spectrum (C) 2D histogram of orientation rays (D) Probability Mass Function (PMF) for the global perspective features (E) Retrieved set (F) Vanishing point candidates.

is shown in Fig.1. First, in place of the traditional noise-sensitive edge-detection step, we analyze the texture by extracting dominant orientations in the image. In contrast to [2, 3] that identify the dominant orientation in each pixel, we extract the dominant orientations only from the patches that demonstrate strong perspective texture information to ensure the algorithm's robustness and efficiency. Second, we compare the global perspective structure of the input image with each image in the database and retrieve the best image candidates to offer their vanishing points as the initial estimates of the input image. Finally, a probabilistic refinement is applied to iteratively improve the estimates.

The rest of this paper is organized as follows: We discuss feature extraction in Section 2 and global perspective structure matching (GPSM) in Section 3. Then, we introduce the probabilistic refinement algorithm in Section 4 to improve the initial estimation. In Section 5, we show the experimental results and conclusions are given in Section 6.

2. GLOBAL PERSPECTIVE STRUCTURE REPRESENTATION

There are several image representation methods widely used for scene matching such as GIST descriptor and SIFT descriptor [5, 6], which group image shape and texture features without any depth information. Nevertheless, they do not emphasize the extraction of image's global perspective structure (e.g., the parallel line structure on road boundaries and lane markers), which is important in structure matching to infer vanishing point candidates. Hence, enhancing the feature extraction method described in [2], the global perspective structure extraction is achieved in two stages.

First, we compute the dominant texture orientations on grid patches of the image and keep those that have strong texture. We divide each image of size 320×240 into 10×10 pixels grid patches. The dominant orientation θ_i of patch i is the direction that describes the strongest local line structure or texture flow. Similar to the approach in [2], we analyze the texture characteristic of the image by convolution with a set of parameterized Gabor filters at 72 orientations to achieve a good angular resolution. To characterize local texture properties at patch i, we examine the filter responses of the Gabor filters at all orientations. The dominant orientation θ_i of the patch i is chosen from the filter orientations which elicits the maximum responses at the position. In contrast to [2] that uses dense texture features from every pixel on the image, we only adopt the orientations from the patches with strong perspective characteristics by thresholding their maximum filter responses. An example of the dominant orientations computed over patches is shown in Fig. 2(A) for a challenging case with noisy lines and traffic.

Second, we render the orientation rays $r_i = (i, \theta_i)$ along with their dominant orientations θ_i on these strong texture patches at the location (x, y) to build the 2D histogram of rays. Each ray will pass through several patches in the image (Fig.2(B)). After rendering all the rays, we formulate a counting function for calculating the number of rays lying on the patch i as follows:

$$C_{x,y}(i) = \sum_{r'_i \in R} count(i, r'_i) \tag{1}$$

where R represents the set of rays from all the strong texture patches. $count(i, r'_i)$ is defined as

$$count(i, r'_i) = \begin{cases} 1, & \text{if ray } r'_i \text{ passes through patch } i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

This leads to the 2D histogram of orientation rays $F = [C_{1,1}(1), C_{1,2}(2), \ldots, C_{(h,w)}(N)]$ over all patches (Fig.2(C)), where N is the number of patches and (h, w) are the image's height and width in terms of the number of patches ($N = w \times h = 768$). We cascade the 2D histogram row-by-row and generate the 1D Probability Mass Function (PMF) as the global perspective feature as shown in Fig.2(D). Since these features only collect the global statistical information of perspective structure, they are invariant to color and illumination changes while being discriminative to the camera's translation and rotation. This makes it able to reveal the image layout for identifying the vanishing point.

3. GLOBAL PERSPECTIVE STRUCTURE MATCHING (GPSM)

Applying a majority voting algorithm as in [2], we can get the patch position with a maximal value as the raw estimate of the vanishing point (indicated by a pink arrow in Fig.2(C)), which, unfortunately, is not the correct estimate (indicated by a black arrow in Fig.2(C)) due to the effects of noisy texture and traffic in a complex scene. In our method, we apply GPSM to retrieve a set of images that closely match the scene perspective and geometrical layout of the input image. After feature extraction, we search the K nearest neighbors in feature space based on the weighted L2-norm distance $DF = \sum_{i=1}^{N} f_i^q (f_i^q - f_i^d)^2$ between features, where f_i^d is the ith feature value of the image in the database, and f_i^q is the ith

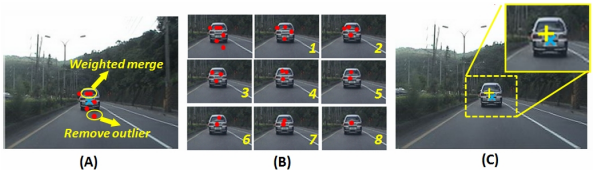

Fig. 3: Iterations of PR. (A) The vanishing point candidates (red dots) and their weighted average (blue cross). (B) 8 PR Iterations, the vanishing point candidates are converging. (C) Comparison between the result by a weighted average (blue cross) and the result using PR algorithm (yellow cross).

feature value of the input image, which also serves as the ith weight of that feature in distance calculation.

Fig.2(E) shows the retrieved sets. Note that retrieved images only match the geometric perspective structure of the input image, but the scene contents are not necessarily similar. The labeled vanishing points of the retrieved set are treated as the vanishing point candidates of the input image (Fig.2(F)).

4. PROBABILISTIC REFINEMENT

Since the neighbor with smaller distance in feature space is more similar to the input image in the global perspective structure, its corresponding labeled vanishing point candidate should have more importance in the initialization of probabilistic refinement. Therefore, we define the prior probability $P_0(k)$ of each vanishing point candidate v_k^0 to be the vanishing point of the input image based on its feature distance DF_k between retrieved images and the input image:$P_0(k) = (1/DF_k)/(\sum_{k=1}^{K} 1/DF_k)$.

The GPSM candidates give rough indication of the vanishing point position of an input image. We can find a vanishing point of the input image through a weight average of all vanishing point candidates using $P_0(k)$ as weights for simplicity. However, possible outlier vanishing point candidates (illustrated in Fig.3(A)) will negatively influence such an average. Therefore, in our method, we refine the initial estimate from GPSM candidates by adopting an iterative probabilistic refinement method, whose task is to find the most plausible vanishing point candidates supported by the most reliable line segments to achieve more precise vanishing point estimation of the input image.

4.1. Vanishing Point Model

Since we have obtained the vanishing point candidates and line segments (orientation rays) from the prior-based GPSM, the problem now is to build a probabilistic model to iteratively estimate parameter representing the location of vanishing points v_k^t at iteration t.

The likelihood function $L(v_k^t; l_i, k)$ can be formulated in terms of their priors $P_t(k)$ and the conditional probability $P_t(l_i|k)$ with respect to the measured line segments l_i: $L(v_k^t; l_i, k) = \prod_{i=1}^{I} P_t(k)P_t(l_i|k)$.

In the ideal case, the supported line segments will pass through the vanishing point. Therefore, we model the offset $d(v_k^t, l_i)$ between line segment l_i (parameterized to a general form of $\alpha_i x + \beta_i y + \gamma_i = 0$) and vanishing point v_k^t with position $v_k^t = (x_k^t, y_k^t)$ by a normal distribution with zero mean and a small variance (e.g., $\sigma^2 = 2$) in the image: $N(d(v_k^t, l_i)|0, \sigma^2) \propto exp(-(d^2(v_k^t, l_i))/(2\sigma^2))$, where the offset $d(v_k^t, l_i)$ defined as: $d(v_k^t, l_i) = (\alpha_i x_k^t + \beta_i y_k^t + \gamma_i)/(\sqrt{\alpha_i^2 + \beta_i^2})$. In addition, we weight the line segment l_i via its normalized sum of maximum responses m_i within the patch i: $P(m_i) = m_i/\sum_{i=1}^{I} m_i$, where I represents the number of rays in the orientation ray set R. Therefore, the conditional probability is computed as: $P_t(l_i|v_k) \propto \exp(-(d^2(v_k^t, l_i))/(2\sigma^2))P(m_i)$

4.2. Expectation Step

Given the current estimate v_k^t, the conditional distribution of vanishing point candidates v_k over line segments l_i: $P_t(k|l_i)$ is determined by Bayes theorem. Then, the Expectation step results in the expected log-likelihood: $Q(v_k^{t+1}|v_k^t) = E[\log L(v_k^t; l_i, k)] = \sum_k \sum_i P_t(k|l_i) \log(P_t(l_i|k)P_t(k))$.

4.3. Maximization Step

By maximizing $Q(v_k^{t+1}|v_k^t)$ obtained in the previous Expectation step, we first re-estimate the prior probabilities that $P_{t+1}(k) = \arg\max_{P_t(k)} Q(v_k^{t+1}|v_k^t) = 1/I \sum_{i=1}^{I} P_t(k|l_i)$. Then, the new estimated location is also calculated by this maximization separately over each vanishing point candidate, i.e., $v_k^{t+1} = \arg\max_{v_k^t} \sum_{i=1}^{I} P_t(k|l_i) \log P_t(l_i|k)$. This is equivalent to a quadratic weighted least-squares problem of the form with $P_t(k|l_i)$ as weight: $v_k^{t+1} = \arg\min_{v_k^t} \sum_{i=1}^{I} P_t(k|l_i) d^2(v_k^t, l_i)$

4.4. Merging and Removal Step

Following each iteration, we employ a greedy search algorithm to find all distances between two vanishing point candidates. When the distance of two candidates is less than a threshold t_{min} ($t_{min} = 5$ pixels)), we merge them using a weighted average into a new vanishing point candidate with their priors $P_t(k)$ as the weights. When the distances of one vanishing point candidate to all other candidates are larger than threshold t_{max} ($t_{max} = 20$ pixels), we remove this candidate as an outlier. Moreover, we will remove the outlier line segments when all the distances between vanishing point candidates and a line segment are more than 3σ as defined above in each iteration. Finally, the iterative probabilistic refinement algorithm runs until convergence to one vanishing point as shown in Fig. 3.

5. EXPERIMENTS

In this section, we show the qualitative and quantitative results of our proposed method. The database has 5500 images sampled at 5 frame per second from the videos covering different road scene scenarios. The collected videos with size 320×240 provide a diversity of image perspectives and their corresponding vanishing point positions (red dots) as illustrated in Fig. 4. We extracted the global perspective structure

Fig. 4: Sample images from image database.

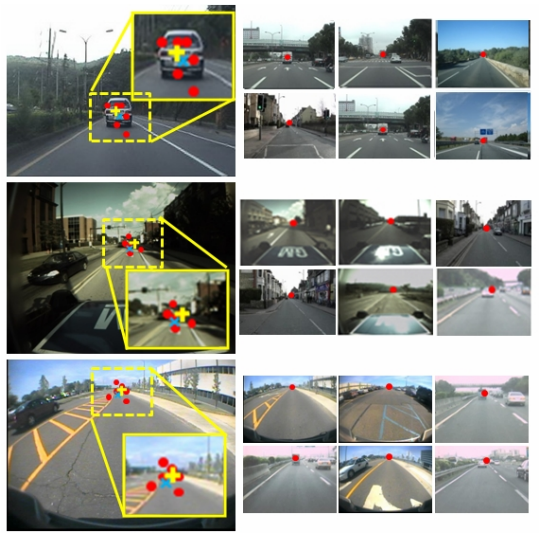

Fig. 5: Experimental results. Left: query images. Right: best matched images. (Red dot: GPSM result. Blue cross: GPSM + WA result. Yellow cross: GPSM + PR results.)

Fig. 6: GPSM + PR results. Vanishing point estimates are shown as red dots, with Gaussian fits of a set of human responses marked with pink ellipses.

features from each image and saved them along with manually labeled vanishing point of each image in the database. The test set comprises 300 images also from the same video sequences, but from different road scene scenarios.

Fig.5 demonstrates estimation results [1] . For each query image, we find the best 6 matched candidates from the database with the closest global perspective structures shown on the right. The vanishing point candidates from neighbors are refined and merged to the final estimate (yellow cross) by probabilistic refinement (PR). Independent of the context of images, our method matches the structures from various scenarios well and gives reasonable positions of vanishing point, even in the complex road scenes such as urban scenario, traffic scenario and distortion scenario, which are difficult to solve by previous methods [1, 2, 3].

To assess the algorithm's performance, we evaluate the proposed vanishing point estimation by comparing the results to human perception, as shown in Fig.6. We indicate the distribution of human choices (10 trained persons for each image) with pink 3σ error ellipses. The mean position difference at the 320×240 scale between our algorithm's estimates and the human choices is 3.5 pixels horizontally and 3.7 pixels vertically, compared to [2] which yields differences of 6.4 pixels horizontally and 6.8 pixels vertically.

Table 1 details the performance of different methods in various scenarios. We compare our proposed GPSM with weighted average and GPSM with PR to the Rasmussen's work in [2]. The proposed methods are slightly better than

[1] More results are available on `http://www.ece.cmu.edu/~qwu/Research/Vanishing/`

Table 1: Comparison of Rasmussen's method and the proposed methods via mean position difference between algorithm's estimates and human choices.

Pos. Diff.	Rasmussen's		GPSM+WA		GPSM+PR	
Mean (STD)	W/O traffic	W/ traffic	W/O traffic	W/ traffic	W/O traffic	W/ traffic
Urban	H:4.8 V:5.1	H:5.6 V:7.2	H:4.2 V:4.1	H:4.6 V:4.8	H:3.4 V:3.5	H:3.5 V:3.6
Highway	H:4.1 V:3.8	H:4.9 V:5.4	H:3.8 V:3.7	H:4.5 V:4.3	H:3.2 V:2.8	H:3.2 V:3.4
Country Side	H:4.2 V:4.1	H:5.8 V:5.6	H:3.8 V:3.9	H:4.4 V:4.6	H:3.2 V:3.2	H:3.4 V:3.4
Distorted	H:7.3 V:7.7	H:9.3 V:11.3	H:5.9 V:5.6	H:6.9 V:6.8	H:5.4 V:5.3	H:5.7 V:6.2
Overall	**H:6.4(5.4) V:6.8(5.6)**		**H:4.4(3.6) V:4.6(3.8)**		**H:3.5(2.7) V:3.7(2.5)**	

baseline method in urban and highway scenarios without traffic since perspective texture in those scenarios dominates the scene and is sufficient to conduct a simple voting. However, adding the traffics or in complicated scenarios, our prior-based GPSM approaches greatly outperform the baseline method [2]. Especially, we perform fairly well in handling distorted images, where it is difficult to estimate the vanishing point via traditional texture voting and edge-based methods.

6. CONCLUSIONS

We presented a prior-based framework which enabled us to estimate image vanishing point under a wide range of conditions. By extracting features from images with perspective texture information and efficiently exploiting similar structural regularities from the images in the database with pre-labeled vanishing points, we deduce the initial guess of vanishing point positions of input images using GPSM. Probabilistic refinement (PR) algorithm is adopted to optimize results iteratively and to improve the estimation accuracy. We compared the proposed method to a state-of-the-art method and observed that our method provides better accuracy and reliability.

7. REFERENCES

[1] T. Suttorp and T. Bucher, "Robust vanishing point estimation for driver assistance," *IEEE intelligent Transportation Systems Conference*, 2006.

[2] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," *Computer Vision and Pattern Recognition*, 2004.

[3] C. Rasmussen and T. Korah, "On-vehicle and aerial texture analysis for vision-based desert road following," *Computer Vision and Pattern Recognition*, 2005.

[4] A. Gallagher, "A ground truth based vanishing point detection algorithm," *Pattern Recognition*, 2002.

[5] J. Hays and A. Efros, "Scene completion using millions of photographs," *SIGGRAPH*, 2007.

[6] W.T. Freeman A. Torralba, K.P. Murphy and M.A. Rubin, "Context-based vision system for place and object recognition," *International Conference on Computer Vision*, 2003.