

# Real-time Vanishing Point Detector Integrating Under-parameterized RANSAC and Hough Transform

Jianping Wu, Liang Zhang, Ye Liu, Ke Chen  
Suzhou Vocational University  
Suzhou, China 215104

{wujp, zhangl, liuye, cke}@jssvc.edu.cn

## Abstract

*We propose a novel approach that integrates under-parameterized RANSAC (UPRANSAC) with Hough Transform to detect vanishing points (VPs) from un-calibrated monocular images. In our algorithm, the UPRANSAC chooses one hypothetical inlier in a sample set to find a portion of the VP's degrees of freedom, which is followed by a highly reliable brute-force voting scheme (1-D Hough Transform) to find the VP's remaining degrees of freedom along the extension line of the hypothetical inlier. Our approach is able to sequentially find a series of VPs by repeatedly removing inliers of any detected VPs from minimal sample sets until the stop criterion is reached. Compared to traditional RANSAC that selects 2 edges as a hypothetical inlier pair to fit a model of VP hypothesis and requires hitting a pair of inliers, the UPRANSAC has a higher likelihood to hit one inlier and is more reliable in VP detection. Meanwhile, the tremendously scaled-down voting space with the requirement of only 1 parameter for processing significantly increased the performance efficiency of Hough Transform in our scheme. Testing results with well-known benchmark datasets show that the detection accuracies of our approach were higher or on par with the SOTA while running in deeply real-time zone.*

## 1. Introduction

A set of parallel lines in 3-D space converge in image plane to a common point called vanishing point (VP) which is equivalent to the infinity of the set of parallel lines in the original world frame. VP contains important information for perspective projection transformation. The coordinates of a VP fully determine the direction in 3-D space of the set of parallel lines that converge to the VP. Moreover, VPs formed by multiple sets of co-plane parallel lines are collinear. The line that passes these VPs is called a vanishing line, which solely determines the plane containing the

very sets of parallel lines. A triplet of mutually orthogonal VPs can be used to derive the internal as well as external parameters of the camera [6]. A special and also important type of vanishing line is the horizon line, a virtual line passing a set of horizontal vanishing points. In man-made environment, a type of vanishing point called zenith (or nadir when the position is below the image center) is formed by lines perpendicular to horizontal plane in the real world .

VP detection is of relevance in camera calibration ([6], [7], [18], and [1]), 3-D reconstruction ([25], [37]), virtual reality [30], robot navigation [15] and scene tracking [8] and vehicle automatic driving [16]. Consequently, VP detection is an important research topic in computer vision. The location of a VP in the image plane is determined by relative positions and directions of a set of parallel lines displayed in the camera frame. The process of detecting a VP is essentially to search in the image plane for the common intersection of a set of VP's inlier edges (line segments) that are projections of the set of parallel line segments in the original 3-D space. Due to background noise and image distortion, inlier edges of a valid VP generally do not exactly converge to a single point and instead they intersect with each other in a bounded region, whose centroid is regarded as a VP nonetheless.

### 1.1. Related Works

Having long been considered an important research topic in computer vision, VP detection remains a challenging issue that is still far from being perfectly resolved.

Robust in noise environment, Hough Transform (HT) [12] was among the first to be used in VP detection. The obstacle of applying HT to VP detection is that HT only works in a bounded parameter space whereas the image plane is unbounded. To overcome this obstacle, [28] divided the image plane into 3 bounded subspaces in which the cascaded 2-D Hough Transform was then conducted. For calibrated images with known focal length, Barnard [3] mapped the image plane into the Gaussian sphere and conducted HT in the latter space. Later, improvements on [3] were proposed

by [24]) and [20]. Overall, the accuracy of HT is subject to discretization effect of Hough cells sampled inside the 2-D parameter space and the corresponding computational complexity is generally high in order to maintain satisfactory detection accuracy, which is proportional to the number of cells sampled over the entire 2-D space.

Like HT, RANSAC [11] has also been widely explored for VP detection. Unlike HT, RANSAC does not require the partitioning of the parameter space and can directly work in the image plane. RANSAC is particularly suitable for VP detection in man-made environments where a substantial share of the input edges are inliers of a few and often mutually orthogonal VPs (see [22]). The most promising RANSAC-based scheme was proposed by Tardif [26] which randomly selects up to 500 minimal sample sets of edge pair from a set of input edges and further fits the input edges to the hypothesis of all-inlier pair using J-Linkage to find the pairs with satisfactory consensus. To further improve accuracy of Tardif's algorithm, [27] ran the J-Linkage several times after different random initializations whereas [34] adopted a much larger sample set size. In more limited settings where inliers of the orthogonal VPs are dominant in the input edges, a RANSAC hypothesis set could even be expanded to 3 edges belonging to a triplet of orthogonal VPs [21] or 4 edges belonging to 2 or 3 orthogonal VPs [33]. For calibrated images, [19] proposed to use the focal-length-dependent equivalent sphere, which is similar to Gaussian sphere, to conduct real-time VP detection through RANSAC.

Recently, neural network has been explored for VP detection. Neural network approaches generally favor the data representation on the Gaussian sphere (see [3]), which is a bounded space ideal for neural-network-based VP detection. [35] proposed a scheme that uses global image context extracted via a deep convolutional neural network (CNN) on Gaussian sphere to address horizontal and zenith VP detection. [36] proposed Hough-Transform-based conic convolution operator to evaluate a set of VP candidates sampled from Gaussian sphere. [13] combined CNN and RANSAC that makes use of Gaussian sphere representation arising from an inverse gnomonic projection of edge lines. [14] proposed a neural network conditioned on multi-model-based sequential RANSAC to perform different types of model fitting including VPs estimation, line-fitting, *etc.* [23] combined Hough Transform with neural network to address the issue of VP detection in documental images.

Apart from the aforementioned methods, other approaches have also been proposed for VP detection. Antunes *et al.* [1] proposed a global algorithm using a local message passing approach to solve VP detection as a multi-model fitting problem, which showed comparable detection accuracy to [26]; [17] resorted to PCLines dual spaces [10] to perform VP detection; [4] proposed a mathematically

guaranteed globally optimal manner which inherently enforces the VP orthogonality.

## 1.2. Contribution

In finding 2 degrees of freedom of a VP, the existing RANSAC ([26]) requires 2 hypothetical VP inliers to fit a VP model. For an image whose edge set contains a large number of VP outliers, the number of sample sets required to hit an all-inlier (outlier-free) minimal set with reasonable probability grows exponentially large.

To reduce the number of the sample sets without sacrificing reliability, we propose UPRANSAC in which a sample set only contains a single hypothetical VP inlier (HVPI). With UPRANSAC fixing only part of the degrees of freedom (DOF) of a VP, its remaining DOF is later recovered by a reliable brute-force voting scheme (1-D HT conducted along the extension line of the HVPI). In performing HT, the whole Hough space consists of the extension lines of all of the selected HVPIs with each line being regarded as a Hough subspace. Each subspace is further partitioned into a set of Hough cells of approximately equal probability. After voting, a VP is detected by searching for the HVPI owning the optimal consensus, which is defined as the HVPI whose associated subspace owns the peak cell across the whole Hough space. Our approach detects multiple VPs sequentially by removing HVPIs of the detected VP from the minimal sample sets (and removing from the Hough space the subspaces associated with these HVPIs) and repeating until the stop criteria (*e.g.* the number of detected VPs is above a certain threshold or the vote count of the peak cell across the whole Hough space is below a prescribed threshold) are reached.

So, our major contribution in this paper is proposing a real-time VP detector that integrates RANSAC and HT, 2 traditionally mutually exclusive techniques, with each of them recovering part of a VP's 2 DOFs. This novel approach brings 3 immediate benefits: 1) the number of sample sets needed to fit a VP model is substantially smaller than a fully-parameterized RANSAC (ours: 100 versus [34]: up to 3,000); 2) the Hough space that consists of multiple 1-D subspaces is easier to handle than the 2-D space (usually Gaussian sphere) used by typical HT approaches; 3) Votes are cast by image edges rather than edge pairs (see [3], [20], [24], [28]), and voting is conducted more concisely in 1-D space (extension line of a HVPI) instead of 2-D space, leading to significant drop of computational cost.

As a VP can be anywhere along the extension line of an HVPI, our another contribution is proposing a scheme to map the extension line of an HVPI, an unbounded space apparently not partitionable, into a bounded space (an angle space) which can then be partitioned into a set of cells of approximately the same probability. Our mapping scheme

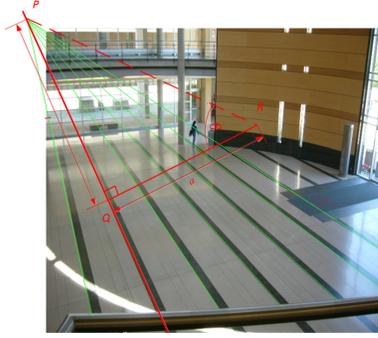


Figure 1. Mapping the coordinate  $t$  of a point  $P$  along a line to the coordinate  $\theta = \tan^{-1} \frac{t}{a}$  in an angular space.

is illustrated in Fig. 1. Assume  $Q$  is the pedal of  $O$  on the extension line of an HVPI (marked with thick red) and  $P$  is the intersection between the extension line of a voting edge and the extension line of the HVPI, we use  $t = QP$  to represent the coordinate of  $P$  on the HVPI line ( $t$  is positive if  $P$  is at the right side of  $OQ$  and negative otherwise). Although  $t$  is unbounded, we propose a scheme that maps  $t$  into  $\vartheta = \tan^{-1} \left( \frac{t}{a} \right)$ , where  $a$  is a parameter dependent on the location and orientation of HVPI line. Apparently,  $\vartheta$  lies in a 1-D bounded space which makes it ideal for HT.

Our VP detection algorithm operates directly in image domain. Since the consistency and distance measures in image domain have clear geometric interpretation, our algorithm is able to achieve detection accuracy and robustness that are comparable to the state of the art; moreover, the computational frugality and space compactness of the 1-D Hough Transform ensures our algorithm's real-time performance.

## 2. Algorithm

Assume a set of  $N$  edges is extracted from an image, the algorithm's pipeline runs through the following 4 steps:

1). In UPRANSAC step, a total of  $n$  sample edges, each of which is regarded as a HVPI, are randomly selected from the  $N$  input edges;

2). For each HVPI, a Hough subspace is built along its extension line through partitioning the line into a set of  $m$  cells of approximately equal probability;

3). Each input edge gets to vote once in each subspace, casting its vote for the cell that the extension line of the voting edge crosses. As HVPI is not allowed to vote in its associated subspace,  $(N - 1)$  votes are cast for  $m$  cells in each subspace;

4). After voting, a multi-round VP detection is conducted with each round detecting a single VP and removing the subspaces associated with those HVPIs which are valid inliers of the VP. The VP detection ends when the stop cri-

teria are reached.

### 2.1. Selection of a set of HVPI edges

Distinct from a full-parameterized RANSAC VP detector such as [26] in which each sample set contains 2 HVPIs to fit a VP model, the UPRANSAC selects a single HVPI as a sample set, in which case a VP is not totally fixed but instead bound to the extension line of the HVPI. This brings an obvious advantage in that a much smaller number of minimal sample sets are required to achieve the same probability of hitting an all-inlier sample set as a full-parameterized RANSAC. For example, assume an image has 1,000 edges, among which 50 are the inliers of a VP. Then in our approach, each sample set has the probability of 0.05 to hit a VP inlier; in contrast, a sample set in a fully-parameterized RANSAC that uses a pair of HVPIs as a sample set has the probability of  $0.05^2 = 0.0025$  to hit an all-inlier set. As the number of meaningful edges (or line segments) in a typical average image is usually less than 1,000, the size of the minimal sample sets needed by UNPANSAC can be as low as 200 input edges to ensure adequate detection reliability (against false negative detection). In practice, a larger size in the minimal sample sets (up to the full set of the input edges) than necessary only marginally improves the detection accuracy.

### 2.2. Construction of Hough space

Our approach's Hough space consists of a set of subspaces, each of which is built along the extension line of an HVPI. To optimize the HT reliability, it is necessary to formulate a scheme to partition each subspace into a set of Hough cells of approximately equal probability. In this way, the votes cast by outliers of a VP tend to be distributed evenly over the cells of the subspace and thus the risk of false positive VP detection is minimized.

For simplicity, the image is first normalized to a unit square with its center fixed at the origin as shown in Fig.2. We use  $l(d, \phi)$  to represent the extension line of an HVPI, where  $d$  is the distance between  $O$  and the HVPI line,  $Q$  is the pedal of perpendicular projection of  $O$  to the HVPI, and  $\phi$  is the level-line angle of  $OQ$ . An image generally contains a number of edges (line segments), whose location and orientation distributions are quite different from uniform distribution. However, for a huge number of images of the same size, by statistics, their edges are expected to be evenly distributed inside the image region and their orientations are also expected to be evenly distributed between  $0$  and  $\pi$ . Collectively, these edges can be modelled by a simple generic random edge  $e(u, v, \theta)$  whose midpoint  $C$  ( $(u, v)$  in Fig. 2) is uniformly distributed inside the image region and whose level-line angle  $\theta$  is also uniformly distributed in the angular space ( $\theta \in (0, \pi)$ ).

The probability density function of  $e(u, v, \theta)$  can be

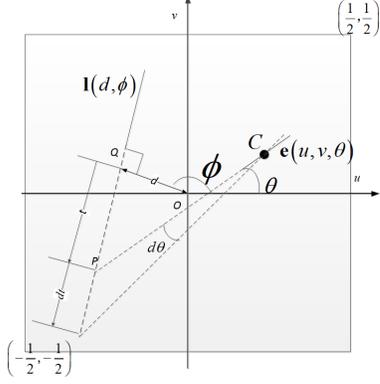


Figure 2. The intersection of the extension line of the random edge  $e(u, v, \theta)$  and the HVPI  $l(d, \phi)$ .

written as:

$$f(u, v, \theta) = \begin{cases} \frac{1}{\pi} & \text{if } u, v \in (-0.5, 0.5) \text{ and } \theta \in (0, \pi), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Our goal is to derive an analytical approximation to divide  $l(d, \phi)$  (the extension line of the HVPI) into a set of 1-D Hough cells so that the extension line of  $e(u, v, \theta)$  is approximately equally likely to cross each cell. Such a model is expected to be applicable to a generic image since  $e(u, v, \theta)$  has similar behavior to the collective effect of the set of edges in a general image.

As shown in Fig. 2, assume the extension line of the random edge  $e(u, v, \theta)$  intersects  $l(d, \phi)$  at  $P$ . Along  $l(d, \phi)$  a new variable  $t = QP$  is introduced. Here  $t$  is assigned positive if  $P$  is at the right side of  $OQ$  and assigned negative otherwise. It can be easily derived that  $P$ 's coordinates are  $(d \cos \phi + t \sin \phi, d \sin \phi - t \cos \phi)$ . Since  $CP$ 's level line angle is  $\theta$ , one has:

$$\tan \theta = \frac{v - d \sin \phi + t \cos \phi}{u - d \cos \phi - t \sin \phi} \quad (2)$$

Since the probability distribution of intersection  $P$ 's coordinate  $t$  along  $l(d, \phi)$  is the cumulative effect of the random edge  $e(u, v, \theta)$  over the edge's entire space, probabilistic density function (PDF) of  $t$  is independent of  $u, v$  and  $\theta$ . So PDF of  $t$  can be defined as  $f_{d, \phi}(t)$ .

To deduce  $f_{d, \phi}(t)$ , let's consider the probability of the intersection point  $P$  falling between  $t$  and  $t + dt$  as shown in Fig. 2, where  $dt$  is an infinitesimal increment of  $t$ . On the one hand, this probability is equal  $f_{d, \phi}(t)|dt|$ ; on the other hand, the probability is also equal to the probability that level-line angle of the random edge falls between  $\theta$  to  $\theta + d\theta$ , with  $t$  and  $\theta$  interrelated via Eq.2. So, we have:

$$f_{d, \phi}(t)|dt| = \int_{-0.5}^{0.5} du \int_{-0.5}^{0.5} dv f(u, v, \theta) |d\theta| \quad (3)$$

$$\text{s.t. } \tan \theta = \frac{v - d \sin \phi + t \cos \phi}{u - d \cos \phi - t \sin \phi}$$

where s.t. is the abbreviation of "subject to". Eq.(3) can be solved by moving  $|dt|$  to the right side:

$$f_{d, \phi}(t) = \int_{-0.5}^{0.5} du \int_{-0.5}^{0.5} dv f(u, v, \theta) \left| \frac{d\theta}{dt} \right| \quad (4)$$

$$\text{s.t. } \tan \theta = \frac{v - d \sin \phi + t \cos \phi}{u - d \cos \phi - t \sin \phi}$$

Combining Eq.1 and Eq.4, we have:

$$f_{d, \phi}(t) = \frac{1}{\pi} \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} \frac{|u \cos \phi + v \sin \phi - d| dudv}{(u - d \cos \phi - t \sin \phi)^2 + (v - d \sin \phi + t \cos \phi)^2} \quad (5)$$

Eq.5 gives the PDF of how the intersection point  $P$  is distributed along  $l(d, \phi)$ . While the PDF itself is apparently not analytically solvable, we can still derive its analytical approximation. To do that, we introduce 2 new terms:  $g(d, \phi) \equiv f_{d, \phi}(0)$  and  $h(d, \phi) = \lim_{t \rightarrow \infty} t^2 f_{d, \phi}(t)$ :

$$g(d, \phi) \equiv \frac{1}{\pi} \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} \frac{|u \cos \phi + v \sin \phi - d| dudv}{(u - d \cos \phi)^2 + (v - d \sin \phi)^2} \quad (6)$$

$$h(d, \phi) \equiv \frac{1}{\pi} \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} |u \cos \phi + v \sin \phi - d| dudv \quad (7)$$

The distributions of  $g(d, \phi)$  and  $h(d, \phi)$  over  $d \in (0, 0.5)$  and  $\phi \in (0, 2\pi)$  are shown in Fig. 3 (a) and Fig. 3 (b), respectively. A new function,  $\hat{f}_{d, \phi}(t)$  is introduced here to approximate  $f_{d, \phi}(t)$ :

$$\hat{f}_{d, \phi}(t) = \frac{g(d, \phi)}{1 + \frac{g(d, \phi)}{h(d, \phi)} t^2} \quad (8)$$

Since  $\frac{\hat{f}_{d, \phi}(0)}{f_{d, \phi}(0)} = 1$  and  $\lim_{t \rightarrow \infty} \frac{\hat{f}_{d, \phi}(t)}{f_{d, \phi}(t)} = 1$  and both  $\hat{f}_{d, \phi}(t)$  and  $f_{d, \phi}(t)$  are monotonously decreasing functions of  $|t|$ ,  $\hat{f}_{d, \phi}(t)$  is expected to be a good approximation of  $f_{d, \phi}(t)$ . As  $g(d, \phi)$  and  $h(d, \phi)$  are still not in analytical forms, we introduce their respective empirical approximate solution:

$$g(d, \phi) \approx 0.9 - 0.9d^2 \quad (9)$$

$$h(d, \phi) \approx 0.111 + 0.155d^2 \quad (10)$$

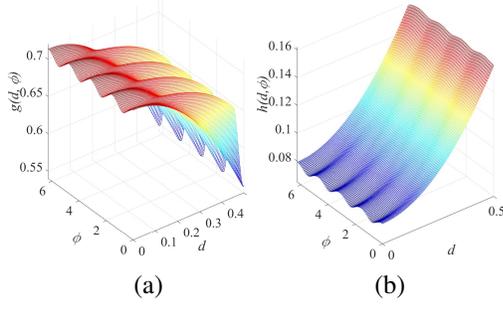


Figure 3. The distribution of  $g(d, \phi)$  (a) and  $h(d, \phi)$ (b) in terms of  $d$  and  $\phi$ .

In above approximate solution, the variable  $\phi$  is ignored since its effect on both  $g(d, \phi)$  and  $h(d, \phi)$  is insignificant in comparison with  $d$ . As a perfect PDF has the property of  $\int_{-\infty}^{+\infty} \hat{f}_{d,\phi}(t)dt \equiv 1$ , the above analytical approximations satisfy  $|\int_{-\infty}^{+\infty} \hat{f}_{d,\phi}(t)dt - 1| \leq 0.01$  for  $d \leq 0.6$ , which makes  $\hat{f}_{d,\phi}(t)$  a satisfactory probabilistic density function for a HVPI edge that is no more than 0.6 times of the normalized image width(or height) away from the image center.

Since  $\hat{f}_{d,\phi}(t)$  is still a function defined over an unbounded 1-D space( $t \in (-\infty, +\infty)$ ), we introduce a new variable  $\vartheta$ :

$$\vartheta = \tan^{-1} \left( \frac{t}{\sqrt{\frac{h(d,\phi)}{g(d,\phi)}}} \right) \quad (11)$$

Since  $\lim_{t \rightarrow \infty} \vartheta = \frac{\pi}{2}$ , obviously  $\vartheta$  is defined in a bounded space ( $\vartheta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ ), with  $\vartheta = \pm \frac{\pi}{2}$  corresponding specifically to cases where the random edge  $e(u, v, \theta)$ (voting edge in Fig. 4) is parallel to the HVPI line  $l(d, \phi)$ . As shown in Fig. 4, assume a voting edge intersects the  $l(d, \phi)$  at point  $P$  which is  $t$  away from the point  $Q$  (the pedal of  $O$  on  $l(d, \phi)$ ),  $t$  can be mapped into an angular variable  $\vartheta = \angle PRQ$  with  $R$  being the reference point along the extension line of  $QO$  and satisfying  $QR = \sqrt{\frac{h(d,\phi)}{g(d,\phi)}}$ .

Corresponding to the PDF  $\hat{f}_{d,\phi}(t)$  of point  $P$  in  $t$  space, we also define the PDF of point  $P$  in  $\vartheta$  space as  $\hat{f}_{d,\phi}(\vartheta)$ . The 2 PDFs are interrelated in the form of:

$$\hat{f}_{d,\phi}(\vartheta)d\vartheta = \hat{f}_{d,\phi}(t)dt \quad (12)$$

Eq.11 yields:

$$\hat{f}_{d,\phi}(\vartheta) = \frac{\hat{f}_{d,\phi}(t)}{\frac{d\vartheta}{dt}} \quad (13)$$

Combining Eq.8, Eq.11 and Eq.13 yields:

$$\hat{f}_{d,\phi}(\vartheta) = \sqrt{g(d, \phi)h(d, \phi)} \quad (14)$$

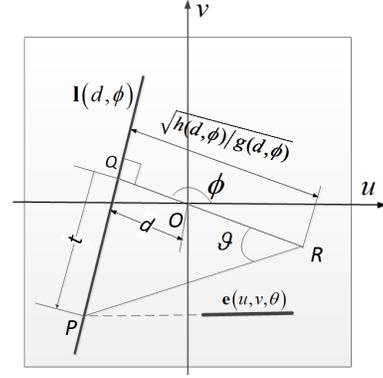


Figure 4. Mapping from  $t$  space to  $\vartheta$  space.

Eq.14 defines a uniform distribution of the point  $P$  in the  $\vartheta$  space. Since partitioning the  $\vartheta$  space into a set of equal-sized cells leads to each cell having the same probability of capturing  $P$ ,  $\vartheta$  space is ideal for HT.

In practice, HT is conducted in  $t$  space but implicitly invoking  $\vartheta$  as far as space partitioning is concerned. Assume the HVPI line is partitioned into a set of  $m$  (an even integer) Hough cells (via  $\vartheta$ ), numbered  $(0, 1, 2, \dots, m-1)$  and a voting edge intersects the HVPI line at  $t$ , then the sequence number  $k$  of the cell that the voting edge casts its vote for is:

$$k = \text{round} \left[ \frac{m}{2} + \frac{m}{\pi} \tan^{-1} \left( \frac{t}{\sqrt{\frac{h(d,\phi)}{g(d,\phi)}}} \right) \right] \quad (15)$$

where the analytical approximations of  $g(d, \phi)$  and  $h(d, \phi)$  are given by Eq.9 and Eq.10 respectively, and "round" is the function that produces the nearest integer around the parameter in the brackets.

### 2.3. HVPI-based Voting

Assume a total of  $N$  edges have been extracted from an image, a subset which contains  $n$  edges is selected in the minimal HVPI sets. The Hough space consists of  $n$  subspaces with each subspace built along the extension line of an individual HVPI and containing  $m$  Hough cells.

To begin with, each Hough cell is initialized to 0 (containing 0 votes). Then, each of the  $N$  input edges casts one vote in each subspace, voting for the cell whose sequence number is derived through Eq.15. In the case where the voting edge is parallel to the HVPI, the vote goes to the first Hough cell ( $k = 0$ ). To ensure the fairness of voting process, a HVPI edge is not allowed to vote in its associated subspace. Overall,  $(N-1) \times n$  votes are cast. With there being  $m$  Hough cells, each cell receives  $\frac{N-1}{m}$  votes on average. Fig. 5 shows an image's input edges (in yellow), 4 HVPIs (in thick red) and the distributions of votes in their

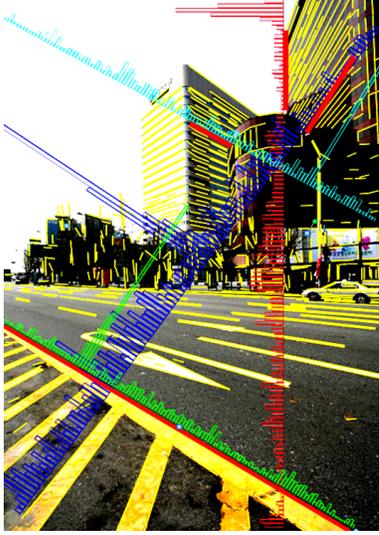


Figure 5. 4 HVPIs and the vote distributions in their respective subspaces.

respective subspaces (in red, green, blue, and cyan). Note that each subspace contains a salient peak which is at least 4 times as high as the average number of votes captured by other Hough cells. It is also observed that each valid region around the peak, which contains those surrounding cells capturing at least half of the votes received by the peak cell, covers no more than 3 cells (In Fig. 5,  $m = 120$ ).

## 2.4. Multi-round VP Detection

The detection stage is a multi-round process with each round consisting primarily of the following 3 steps:

- 1). Search and validate the optimal HVPI. This is the HVPI whose associated subspace contains the peak cell (the cell that captures the most votes across the whole Hough space). If the peak cell's vote count is above a prescribed threshold  $C_{\text{threshold}}$ , a new VP is detected and the optimal HVPI is considered as a valid inlier of a VP;

- 2). Extract the VP's initial estimate and all of its inlier edges, and refine the VP estimate based on the inliers. The VP's initial estimate is set at the midpoint of the peak cell in the optimal HVPI's associated subspace. For each detected VP, we create an inlier set with the optimal HVPI being the first entry. Then those voting edges whose votes fall in the full-width-at-half-maximum region of the peak cell—which covers the cells surrounding the peak with each cell receiving at least half as many votes as the peak cell does—are added into the inlier set of the VP. Please note that in a subspace, the leftmost cell ( $k = 0$ ) and the rightmost cell ( $k = m - 1$ ) are neighboring cells because their respective  $\vartheta$  values are separated by  $\pi/m$ . We use the method that is similar to [35] to refine the VP;

- 3). Update the Hough space based on the VP's inlier set.



Figure 6. The detected VP points and their respective inlier edge sets .

It is assumed that an edge can be an inlier to no more than one VP, so the subspaces of those HVPIs identified as the inliers of the newly detected VP are removed from the current Hough space to prevent duplicate detection; moreover, votes casted by each of the inlier edges of the VP in the current Hough space is discarded to prevent their interference with the following rounds of detection.

The above process runs repeatedly until either the number of detected VPs reaches the prescribed threshold  $N\_MAX_{VP}$  or the votes received by the optimal HVPI's peak cell falls below a prescribed threshold  $C_{\text{threshold}}$ .

Fig. 6 shows 4 sets of VP inliers based on the voting results as shown in Fig. 5. Note only 3 horizontal VPs, which are marked green, blue, and cyan respectively, are shown and their inlier edges are marked with corresponding colors. As expected, the 3 horizontal VPs are colinear and fall on the dashed red line (the horizon).

## 3. Complexity Analysis

The computational cost of our approach primarily comes from 3 sources, voting, VP detection, and VP refining. Here,  $N$ ,  $n$ , and  $m$  are the number of input edges, the number of the HVPIs, and the number of Hough cells of each subspace, respectively. In the implementation,  $n$  and  $m$  are limited with  $n = \min(N, 500)$  and  $m \leq 200$ .

Voting process involves  $(N - 1) \times n$  votes. As  $n \leq 500$ , its computational complexity is  $O(N)$ . VP detection involves at most  $N\_MAX_{VP}$  (6 in our implementation) rounds with each round searching for at most  $m \times n$  cells. With both  $m$  and  $n$  limited, it has the computational complexity of  $O(1)$ . VP refining involves at most  $N\_MAX_{VP}$  VPs, each VP refining involves  $O(N)$  inlier edges, so the total refining cost is  $O(N)$ .

Overall, the algorithm's computational cost is  $O(N)$ , where  $N$  is the number of input edges.

As far as space complexity is concerned, since Hough space involves  $n$  subspaces, each of which contains  $m$  cells



Figure 7. Exemplary VP detection results using our approach. 1st and 2nd rows show the satisfactory detection and the 3rd row shows the less-than-ideal detection cases which account for less than 7% of total cases. Inlier edges are color-coded based on their respective VP type (zenith: magenta; the 2 dominant horizontal VPs: yellow and cyan). Ground truth VPs and detected VP are displayed as circles and red circles respectively; ground truth horizon lines and detected horizon lines are displayed as green solid lines and red dash lines respectively. Note that some VPs are out of the displayable areas and so only their inliers are shown.

(both  $n$  and  $m$  are limited), our approach has the space complexity of  $O(mn) = O(1)$ .

#### 4. Evaluation

Our VP detection scheme is examined and evaluated against the SOTA in VP detection. To make fair comparison, we used a 6-year-old laptop powered by 2Ghz Intel Core i7 CPU and 4GB of DRAM, similar to hardware setup in [17], to perform the tests. Our algorithm is implemented with C++ language which is comparable to [34], [35] and [29], which were implemented with C language, and [17] which was implemented with Matlab+mex.

We evaluate our algorithm using two benchmark datasets, the York Urban Dataset (YUD, [9]) and the Eurasian Cities Dataset (ECD, [2]). YUD contains 102 calibrated images of 640×480 pixels obeying Manhattan world

assumption [8]. ECD contains 103 non-calibrated images of various widths and heights. ECD contains larger images with more complex scenes than YUD and is thus more challenging and time-consuming than YUD with respect to VP detection.

We used Canny edge detector [5] in combination with [32] to extract image edges because of its speed advantage over [31]. Table 1 lists the parameters and their values that our implementation used for evaluation.

Fig. 7 shows results for our algorithm for some images in YUD and ECD. Qualitatively, the case of total failure was not observed as the zenith VP and at least one of the horizontal VPs were always successfully detected by our approach. The first and the second row in Fig. 7 displays the cases where horizon detection error is less than 0.01 (92 among a total of 205 images) with detected horizon line (in red dash line) being very close to the ground truth (solid

Name	Usage	Magnitude(s)
$n$	2.1, 2.2, 3	$\min(500, N)$
$m$	2.1, 2.2, 3	158 (YUD) or 180 (ECD)
$N\_MAX_{VP}$	2.3, 3	6
$C_{threshold}$	2.4	$\max\left(5, \frac{3N}{m}\right)$

Table 1. Algorithm parameters used.

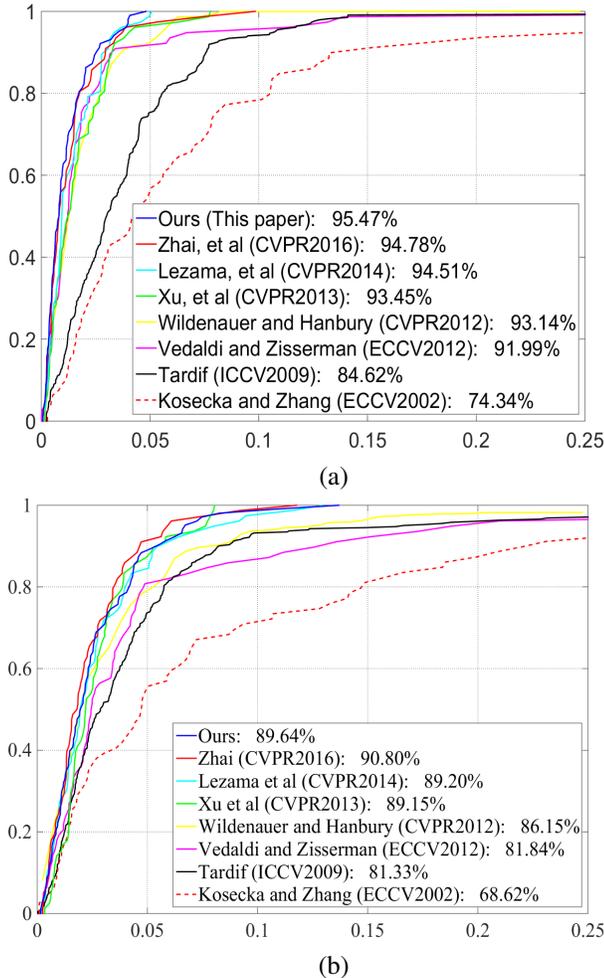


Figure 8. Quantitative evaluation of the horizon line estimation. The horizontal axis represents the horizon line detection error. The vertical axis represents the ratio of images with horizon line error lower than the corresponding abscissa. The AUC for each curve is shown in the legend. For additional details see Sec

green line), and the third row shows the cases where horizon detection error is higher than 0.05, indicated by the significantly wider gap between detected horizon and ground truth.

It shows that our approach is generally able to robustly handle various types of noises, including irregular building contour (3rd and 4th images in 1st row and 1st and 3rd im-

ages in 2nd row), stair handle (2nd image in 1st row), lane markers and other road surface objects (2nd, 3rd, and 4th images in 2nd row). The partially failed detection, which is defined as horizon detection error greater than 0.05, makes up 6.3% of the all tested images. The primary cause of the partial VP detection failure is clustering of unrelated edges into a set of inliers for the same VP. Those edges prone to VP detection error usually include vehicle contours (1st and 3rd images in the 3rd row), electric wires (the 3rd and 4th images in the 3rd row), edges belonging to different sets of parallel lines (the 2nd and 4th images in the 3rd row).

Following previous research works, we use the horizon detection error as a benchmark of VP detection accuracy. Horizon detection error is defined as the maximum vertical distance inside the image between the detected horizon line and the ground truth horizon line normalized by the image height. Following protocol used by [34], [17], [35], [29], the cumulative histogram of these errors is shown and the area under the curve (AUC) is reported as well.

Fig. 8(a) shows the accuracy comparison of the 8 competing algorithms using YUD dataset. Our approach achieves the best AUC at 95.47%, which is 0.69% better than the second best at 94.78% achieved by [35]. Fig. 8(b) shows the detection errors for the 8 algorithm on ECD data, in which our approach achieves the second best AUC of 89.64%, which is about 1.16% lower than the best AUC (90.80%) achieved by [35] but 0.44% better than the 3rd best AUC (89.20%) achieved by [17].

For a YUD image, our approach’s average end-to-end processing time—including image loading, edge detection and VP detection—is 43 milliseconds, which drops to 9.3 milliseconds excluding image loading and edge detection. In comparison, [35] and [17] required 1 second and 30 seconds, respectively, on average to process a YUD image.

## 5. Conclusion

We presented a novel VP detection scheme that demonstrates competitive accuracy with the state of the art at real-time processing efficiency. The further work will be in adding neural network module to recognize edges belonging to unrelated objects to reduce partial failed detection error.

## 6. Acknowledgment

The research has been funded in part by National Science Foundation of China (61672372) and Suzhou Industrial Application Foundation (SG1819).

## References

- [1] Michel Antunes and Joao P Barreto. A global approach for the detection of vanishing points and mutually orthogonal vanishing directions. In *Proceedings of the IEEE Conference*

- on *Computer Vision and Pattern Recognition*, pages 1336–1343, 2013. 1, 2
- [2] Olga Barinova, Victor Lempitsky, Elena Tretyak, and Pushmeet Kohli. Geometric image parsing in man-made environments. In *European conference on computer vision*, pages 57–70. Springer, 2010. 7
- [3] Stephen T Barnard. Interpreting perspective images. *Artificial intelligence*, 21(4):435–462, 1983. 1, 2
- [4] Jean-Charles Bazin, Yongduek Seo, Cédric Demonceaux, Pascal Vasseur, Katsushi Ikeuchi, Inso Kweon, and Marc Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645. IEEE, 2012. 2
- [5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 7
- [6] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International journal of computer vision*, 4(2):127–139, 1990. 1
- [7] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 941–947. IEEE, 1999. 1
- [8] James M Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *NIPS*, volume 2, page 3, 2000. 1, 7
- [9] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*, pages 197–210. Springer, 2008. 7
- [10] Markéta Dubská, Adam Herout, and Jiří Havel. Pclines—line detection using parallel coordinates. In *CVPR 2011*, pages 1489–1494. IEEE, 2011. 2
- [11] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2
- [12] Paul VC Hough. Machine analysis of bubble chamber pictures. In *Proc. of the International Conference on High Energy Accelerators and Instrumentation, Sept. 1959*, pages 554–556, 1959. 1
- [13] Florian Kluger, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Deep learning for vanishing point detection using an inverse gnomonic projection. In *German Conference on Pattern Recognition*, pages 17–28. Springer, 2017. 2
- [14] Florian Kluger, Eric Brachmann, Hanno Ackermann, Carsten Rother, Michael Ying Yang, and Bodo Rosenhahn. Consac: Robust multi-model fitting by conditional sample consensus. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4634–4643, 2020. 2
- [15] Till Kroeger, Dengxin Dai, and Luc Van Gool. Joint vanishing point extraction and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2449–2457, 2015. 1
- [16] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. Vpnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1947–1955, 2017. 1
- [17] José Lezama, Rafael Grompone von Gioi, Gregory Randall, and Jean-Michel Morel. Finding vanishing points via point alignments in image primal and dual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 509–515, 2014. 2, 7, 8
- [18] David Liebowitz and Andrew Zisserman. Combining scene and auto-calibration constraints. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 293–300. IEEE, 1999. 1
- [19] Xiaohu Lu, Jian Yaoy, Haoang Li, Yahui Liu, and Xiaofeng Zhang. 2-line exhaustive searching for real-time vanishing point estimation in manhattan world. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 345–353. IEEE, 2017. 2
- [20] Evelyne Lutton, Henri Maitre, and Jaime Lopez-Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE transactions on pattern analysis and machine intelligence*, 16(4):430–438, 1994. 2
- [21] Faraz M Mirzaei and Stergios I Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *2011 International Conference on Computer Vision*, pages 2454–2461. IEEE, 2011. 2
- [22] Carsten Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10):647–655, 2002. 2
- [23] Alexander Sheshkus, Anastasia Ingacheva, Vladimir Ar-lazarov, and Dmitry Nikolaev. Houghnet: neural network architecture for vanishing points detection. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 844–849. IEEE, 2019. 2
- [24] Jefferey A Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE transactions on pattern analysis and machine intelligence*, 21(3):282–288, 1999. 2
- [25] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. 2009. 1
- [26] Jean-Philippe Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1250–1257. IEEE, 2009. 2, 3
- [27] Elena Tretyak, Olga Barinova, Pushmeet Kohli, and Victor Lempitsky. Geometric image parsing in man-made environments. *International Journal of Computer Vision*, 97(3):305–321, 2012. 2
- [28] Tinne Tuytelaars, Luc Van Gool, Marc Proesmans, and Theo Moons. The cascaded hough transform as an aid in aerial image interpretation. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 67–72. IEEE, 1998. 1, 2

- [29] Andrea Vedaldi and Andrew Zisserman. Self-similar sketch. In *European Conference on Computer Vision*, pages 87–100. Springer, 2012. [7](#), [8](#)
- [30] Jonathan Ventura and Tobias Höllerer. Structure and motion in urban environments using upright panoramas. *Virtual Reality*, 17(2):147–156, 2013. [1](#)
- [31] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):722–732, 2008. [7](#)
- [32] Geoff AW West and Paul L Rosin. Multistage combined ellipse and line detection. In *BMVC92*, pages 197–206. Springer, 1992. [7](#)
- [33] Horst Wildenauer and Allan Hanbury. Robust camera self-calibration from monocular images of manhattan worlds. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2831–2838. IEEE, 2012. [2](#)
- [34] Yiliang Xu, Sangmin Oh, and Anthony Hoogs. A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1376–1383, 2013. [2](#), [7](#), [8](#)
- [35] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5657–5665, 2016. [2](#), [6](#), [7](#), [8](#)
- [36] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. Neurvps: Neural vanishing point scanning via conic convolution. *arXiv preprint arXiv:1910.06316*, 2019. [2](#)
- [37] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7698–7707, 2019. [1](#)