

2-Line Exhaustive Searching for Real-Time Vanishing Point Estimation in Manhattan World

Anonymous WACV submission

Paper ID 137

Abstract

This paper presents a very simple and efficient algorithm to estimate 1, 2 or 3 orthogonal vanishing point(s) on a calibrated image in Manhattan world. Unlike the traditional methods which apply 1, 3, 4, or 6 line(s) to generate vanishing point hypotheses, we propose to use 2 lines to get the first vanishing point \mathbf{v}_1 , then uniformly take sample of the second vanishing point \mathbf{v}_2 on the great circle of \mathbf{v}_1 on the equivalent sphere, and finally calculate the third vanishing point \mathbf{v}_3 by the cross-product of \mathbf{v}_1 and \mathbf{v}_2 . There are three advantages of the proposed method over traditional multi-line method. First, the 2-line model is much more robust and reliable than the multi-line method, which can be applied in the scene with 1, 2 or 3 orthogonal vanishing point(s). Second, the probability of the 2-line model being formed of inner line segments can be calculated given the noise ratio, which means that the number of iterations can be determined, and thus the estimation of vanishing points can be performed in a very simple exhaustive way instead of the traditional RANSAC method. Third, the real-time performance is achieved by building a polar grid for the line intersection points, which functions as a lookup table for the validation of vanishing point hypotheses. Our algorithm has been validated successfully in the YUD dataset and sets of challenging real images.

1. Introduction

In this paper, we investigate the problem of vanishing point estimation on a calibrated image in Manhattan world. The orthogonal vanishing points provide the informations of the camera orientation, the world structure and the internal parameters of the image, thus it can be used in many applications like single view reconstruction [6], camera self calibration [4], visual navigation [8, 2], and so on. Estimation of vanishing points in the Manhattan world is a tough task because there are two conditions that should be taken into consideration: (1) the global constraint, which means

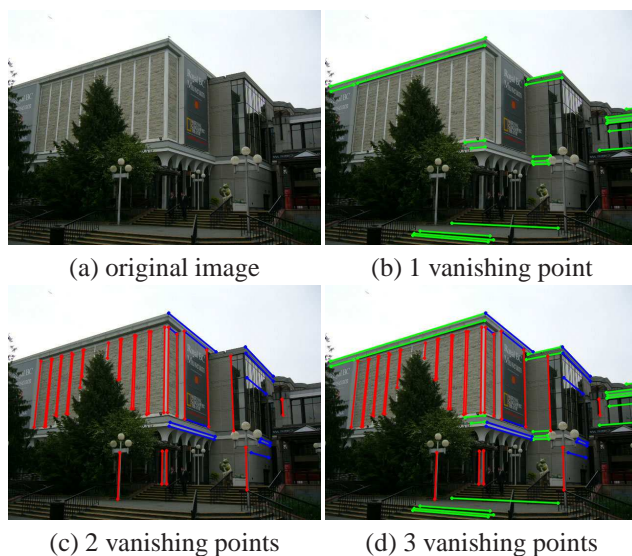


Figure 1. An illustration of vanishing point estimation results of the proposed method in the cases of 1 vanishing point (b), 2 vanishing points (c) and 3 vanishing points (d). The proposed method can estimate 1, 2 or 3 orthogonal vanishing point(s) on a calibrated image in Manhattan world.

that the solution should be global optimal; (2) the orthogonal constraint, which means that the vanishing points should be orthogonal to each other. In the past decades, many algorithms have been proposed to deal with these problems in various view points. In general, these algorithms can be divided into four categories: the exhaustive searching based ones, the expectation-maximization (EM) based ones, the RANSAC based ones, and the optimization based ones.

The exhaustive searching based methods [10, 13] come at the earliest, because they are quite straightforward to take exhaustive samples of the vanishing points and then validate the hypotheses one by one to get the best estimation result. In the work of [13], the equivalent sphere is divided into small accumulator cells and each line segment votes for these cells. Then the cell with the highest vote is chosen as the first vanishing point \mathbf{v}_1 , and the pairs of \mathbf{v}_2 and \mathbf{v}_3 are

exhaustively searched in the remaining cells. The pair with the highest vote is considered as the best estimated vanishing points. However, the computation complexity of the searching step is $\mathcal{O}(n^5)$, which means it can not be applied on some real-time applications. Bazin et al. [2] also used the exhaustive searching strategy, however their method was applied on a navigation system and needs the initial estimations of vanishing points which is provided by other sensors.

The expectation-maximization (EM) based methods [1, 5] perform both line segment classification and vanishing point estimation tasks in an alternative way which finds the best line segment classification given the current estimated vanishing points in the E-step, and estimates the best vanishing points based on the line segment classification in the M-step. The E-step and M-step are performed iteratively until convergence. Antone and Teller [1] proposed to project the line segments onto a cube surface to get straight lines, then apply the Hough Transform, and finally find and validate peaks in the HT space via an EM procedure. Denis et al. [5] developed an approach to independently estimate the vanishing points via the EM procedure and then re-orthogonalize them to fit for the Manhattan world assumption. The well-known shortage of these EM based methods is that they require a precise initial estimation of the vanishing points, which however is not available in most cases. Also, both the global and orthogonal constraints are not well addressed in these EM based methods.

The RANSAC based methods are the most popular ones, which usually define a minimal solution set (MSS) for the estimation of vanishing points and then apply the RANSAC procedure to generate the vanishing point hypotheses iteratively and choose the best one as the final solution. Various models of the MSS have been proposed including the 1-line [3], 3-line [3, 19, 11], 4-line [18, 19] and 6-line [12]. In the work of [3], the horizon plane and horizontal vanishing point are obtained beforehand, then a 1-line MSS is randomly selected to generate the second vanishing point, and the third one can be calculated by the horizontal and the second vanishing points. In the works of [3, 19, 11], they all require a 3-line MSS and a RANSAC procedure to calculate three orthogonal vanishing points, and the only difference between them lies on the mathematical solution to the 3-line MSS problem. In the works of [18, 19], a 4-line MSS was applied to estimate both the vanishing points and the focal length of the image. Rother [12] applied a 6-line MSS to form three pairs of line segments corresponding to three vanishing points. However there is redundancy in this 6-line MSS because a 3-line MSS is just enough to estimate three vanishing points. Those multi-line MSS can attach the orthogonal constraint well, and also the RANSAC procedure can achieve a good balance between the efficiency and the global constraint. However, most of these multi-line MSS will fail when there is only one vanishing point on the

image, and also the RANSAC procedure suffers from the inherent problem of the local minimal solution.

The optimization based methods [7, 16] aim to find a global optimal solution to the problem of the vanishing point estimation in mathematics. Ikeuchi et al. [7] proposed to convert the task of vanishing point estimation into a consensus set maximization problem over the rotation search space, and then solved it by a branch-and-bound procedure based on the Interval Analysis theory. While Tretyak et al. [16] estimated the vanishing points as part of a parsing framework which models the scene as a composition of geometric primitives spanning different layers from low level (edges) through mid-level (lines and vanishing points) to high level (the zenith and the horizon). However, both of these two methods took more than ten seconds to obtain the best solution, which limits their applications on real-time tasks.

There are also some algorithms which solve this problem in other views, for example, Tardif [14] considered the vanishing point estimation as a pattern recognition problem and applied the JLinkage [15] to detect multiple vanishing points. While recently Lezama et al. [9] estimated the Manhattan and non-Manhattan vanishing points via point alignment in both the straight and twisted dual spaces, and their approach achieves a very impressive performance.

In this paper, we present an algorithm which takes advantages of both the MSS and the exhaustive searching strategy to satisfy both the global optimal and orthogonal constraints. To overcome the shortage of the multi-line MSS, we develop a 2-line MSS which is suitable for the estimation of 1, 2 or 3 orthogonal vanishing point(s). To obtain the global optimal estimation result, we apply the exhaustive searching strategy. To accelerate the time-consuming exhaustive searching procedure, we build a polar grid for the line intersection points and convert the exhaustive searching problem into a lookup issue whose complexity is $\mathcal{O}(n)$. The proposed algorithm is very simple and efficient, which contains three steps:

- *Polar grid building*: a polar grid centered on the principle point of the image is built to accumulate the response of line segments for each grid cell.
- *Hypotheses generation*: the 2-line MSS is applied to generate the first vanishing point \mathbf{v}_1 , which is followed by a uniformly sampling of the second vanishing point \mathbf{v}_2 on the great circle of \mathbf{v}_1 on the equivalent sphere, and finally the third vanishing point can be calculated by the cross-product of \mathbf{v}_1 and \mathbf{v}_2 .
- *Hypotheses validation*: all the vanishing point hypotheses obtained in the second step are validated exhaustively via looking up the polar grid build in the first step, and the one with the greatest line segment

response is chosen as the best estimated vanishing points.

2. Algorithm

2.1. Polar Grid Building

The polar grid is built by extending the unit vectors on the equivalent sphere to intersect with the image plane. So first of all, we have a brief introduction to the equivalent sphere applied in this work. Then we will show the details of how to build the polar grid.

Equivalent Sphere: The equivalent sphere is a unit sphere which centers in the focal point of the camera as Figure 2 shows. In this work, the equivalent sphere applies a right-hand coordinate system with the X -axis and Y -axis coinciding with the x -axis and y -axis of the image, respectively, and the Z -axis directs from the focal point of the camera to the principle point of the image. Given the principle point $(x_0, y_0)^T$ and the focal length f of the image, a pixel $(x, y)^T$ on the image can be converted into the equivalent sphere coordinate system via the following formulation:

$$\begin{cases} X = x - x_0, \\ Y = y - y_0, \\ Z = f. \end{cases} \quad (1)$$

For a 3D point \mathbf{P} in the equivalent sphere coordinate system whose coordinates are $(X, Y, Z)^T$, the latitude and longitude (ϕ, λ) of \mathbf{P} is calculated as:

$$\begin{cases} \phi = \arccos(Z / \sqrt{X^2 + Y^2 + Z^2}), \\ \lambda = \arctan2(X, Y) + \pi. \end{cases} \quad (2)$$

Polar Grid: Given a point \mathbf{p} on the image plane, we can obtain its latitude and longitude (ϕ, λ) on the equivalent sphere via Eqs. (1) and (2). The spans of ϕ and λ are $[0, \pi/2]$ and $[0, 2\pi]$, respectively. Thus, the polar grid \mathcal{G} can be built in the following three steps. In the first step, the polar grid \mathcal{G} is zero-initialized with a size of 90×360 with an accuracy of 1° , i.e., $\mathcal{G}(i, j) = 0$ for $i = 1, 2, \dots, 90$ and $j = 1, 2, \dots, 360$. In the second step, for every pair of line segments \mathbf{l}_1 and \mathbf{l}_2 on the image, their intersection point \mathbf{p} is calculated and then the latitude and longitude (ϕ, λ) of \mathbf{p} can be obtained according to Eqs. (1) and (2). Then, we update the corresponding grid cell $\mathcal{G}(\phi_{\text{deg}}, \lambda_{\text{deg}})$ with the following accumulation equation:

$$\mathcal{G}(\phi_{\text{deg}}, \lambda_{\text{deg}}) = \mathcal{G}(\phi_{\text{deg}}, \lambda_{\text{deg}}) + \|\mathbf{l}_1\| \times \|\mathbf{l}_2\| \times \sin(2\theta), \quad (3)$$

where ϕ_{deg} and λ_{deg} stand for the rounds of the degrees corresponding to ϕ and λ , respectively, i.e., $\phi_{\text{deg}} = \lceil \phi \times 180/\pi \rceil$ and $\lambda_{\text{deg}} = \lceil \lambda \times 180/\pi \rceil$, $\|\mathbf{l}\|$ denotes the length of a line segment \mathbf{l} , and θ is the small angle between \mathbf{l}_1 and \mathbf{l}_2 . Eq. (3) is designed to put more weight on the line segment

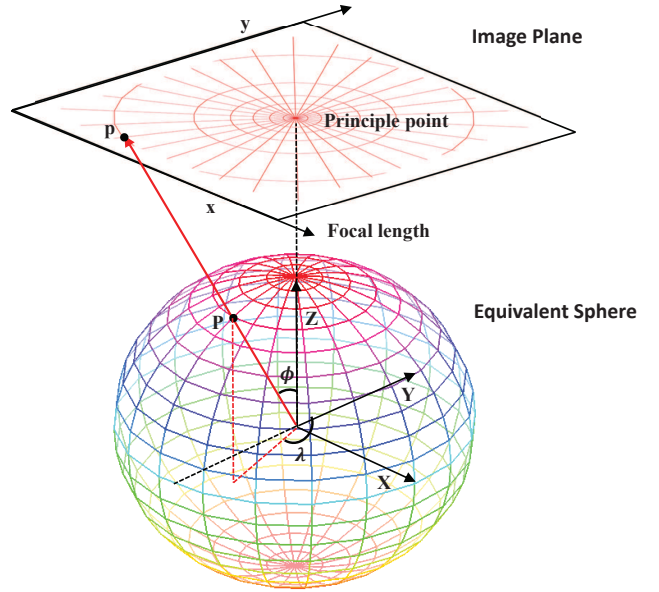


Figure 2. Demonstration of the relationship between the image plane and the equivalent sphere, and the procedure of how to build the polar grid.

pairs with longer length and moderate orientation deviation which means that the angle θ between \mathbf{l}_1 and \mathbf{l}_2 should be neither too small (close to 0°) nor too big (close to 90°). In the final step, a 3×3 Gaussian smoothing filter is applied on the polar grid to get a more robust accumulation result. The result of the polar grid building procedure is a 90×360 grid which records the responses of line segments to each grid cell. Once it has been built, the polar grid can then function as a lookup table, which is very efficient for real-time applications.

2.2. Hypotheses Generation

The exiting vanishing point estimation methods usually use multi-line MSS [3, 19, 11, 18] to generate vanishing point hypotheses. As Wildenauer and Allan [18] have pointed out that the possibility of the multi-line MSS is generally difficult to be calculated directly without a priori knowledge of the population size of line segments corresponding to each vanishing point. Thus in most cases, the iteration number of RANSAC is determined empirically. Unlike those multi-line MSS methods, we propose to use a 2-line MSS to generate the hypotheses only for the first vanishing point and then exhaustively sample for the second vanishing point and generate the third one via the first and second vanishing points. Figure 3 is an illustration of how we generate three orthogonal vanishing points via the following three steps. First, two line segments are randomly selected for intersection to generate the first vanishing point \mathbf{v}_1 . Then, the second vanishing point \mathbf{v}_2 is uniformly sampled on the great circle of \mathbf{v}_1 in the equivalent sphere. Fi-

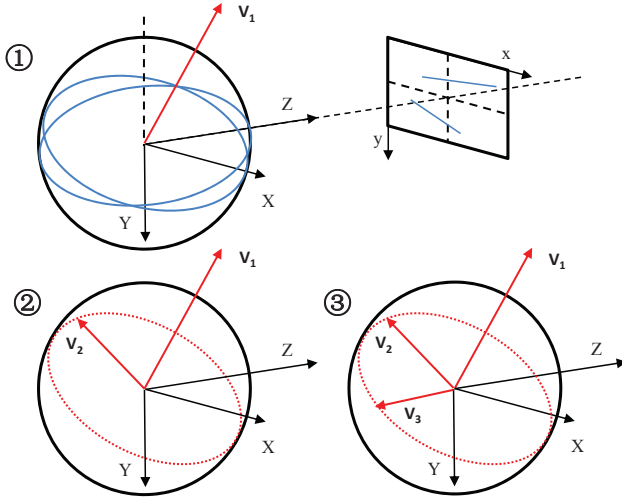


Figure 3. Procedures of how to generate three orthogonal vanishing points.

nally, the third vanishing point \mathbf{v}_3 is calculated by the cross product of \mathbf{v}_1 and \mathbf{v}_2 . In the following parts of this section, we will introduce the details of these three steps.

The First Vanishing Point: First of all, we will discuss the probability of the 2-line MSS. Given an image with N noiseless line segments in total, the number of line segments corresponding to $\mathbf{v}_1, \mathbf{v}_2$ and \mathbf{v}_3 are n_1, n_2 and n_3 , respectively. By randomly selecting two line segments to form a 2-line MSS, the probability P that both these two line segments correspond to the same vanishing point is:

$$P = (C_{n_1}^2 + C_{n_2}^2 + C_{n_3}^2) / C_N^2 \approx (n_1^2 + n_2^2 + n_3^2) / N^2 \in [1/3, 1], \quad (4)$$

which gains the minimal value of $1/3$ when $n_1 = n_2 = n_3 = N/3$.

Then, given a noise ratio of the line segments as 0.5 , the minimal value of P becomes now $1/3 \times 0.5^2 = 1/12$. Under a confidence coefficient of 0.9999 , the number of iteration $\#Its$ needed to obtain at least one inner 2-line MSS is:

$$\#Its = \log(1 - 0.9999) / \log(1 - P) = 105, \quad (5)$$

which means that under a noise ratio of 0.5 and a confidence coefficient of 0.9999 , we need 105 iterations to obtain at least one inner 2-line MSS.

Thus, for the first vanishing point \mathbf{v}_1 , we iteratively and randomly select two line segments and calculate their intersection point, then we convert the intersection point into an unit vector on the equivalent sphere according to Eqs. (1) and (2). In this way, we obtain 105 hypotheses of \mathbf{v}_1 .

The Second Vanishing Point: Considering the orthogonal constraint, given the first vanishing point $\mathbf{v}_1 =$

$(X_1, Y_1, Z_1)^\top$, the second vanishing point must lie on the great circle of \mathbf{v}_1 in the equivalent sphere as shown Figure 3. Thus we uniformly sample 360 hypotheses for the second vanishing point on this circle with an accuracy of 1° , which is performed as follows. First, the longitude span $[0, 2\pi]$ is uniformly divided into 360 fractions, each corresponds to the longitude λ of a hypothesis of \mathbf{v}_2 . Then, for the i -th hypothesis whose longitude is $i \times 2\pi/360$, its latitude ϕ can be calculated via the following constraint equations:

$$\begin{cases} X_2 = \sin(\phi) \times \sin(\lambda) \\ Y_2 = \sin(\phi) \times \cos(\lambda) \\ Z_2 = \cos(\phi), \end{cases} \quad (6)$$

and

$$X_1 \times X_2 + Y_1 \times Y_2 + Z_1 \times Z_2 = 0. \quad (7)$$

Finally, with both the values of ϕ and λ , the second vanishing point $\mathbf{v}_2 = (X_2, Y_2, Z_2)$ can be obtained via Eq. (6).

The Third Vanishing Point: Given the first and the second vanishing points \mathbf{v}_1 and \mathbf{v}_2 , the third vanishing point can be obtained as $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$.

As a summary, after the procedure of hypotheses generation, there will be totally $105 \times 360 = 37800$ hypotheses for three orthogonal vanishing points, which will then be validated to choose the best ones.

2.3. Hypotheses Validation

The final step of the proposed method is the validation of the vanishing point hypotheses. The aim of this procedure is to find out the one with the greatest line segments response among all the 37800 hypotheses. Calculating the line segments response for each hypothesis is time-consuming, however, the polar grid introduced in Section 2.1 can convert this problem into a lookup issue whose complexity is $\mathcal{O}(n)$. The hypotheses validation is performed as follows. For each hypothesis, we calculate the longitudes and latitudes (λ_1, ϕ_1) , (λ_2, ϕ_2) and (λ_3, ϕ_3) via Eq. (2) for $\mathbf{v}_1, \mathbf{v}_2$ and \mathbf{v}_3 , respectively. Then we search for the polar grid, and set $\mathcal{G}([\phi_1 \times \frac{180}{\pi}, [\lambda_1 \times \frac{180}{\pi}]] + \mathcal{G}([\phi_2 \times \frac{180}{\pi}, [\lambda_2 \times \frac{180}{\pi}]] + \mathcal{G}([\phi_3 \times \frac{180}{\pi}, [\lambda_3 \times \frac{180}{\pi}]])$ as the line segments response of this hypothesis. Finally, the one with the greatest response is chosen as the best estimated vanishing points.

2.4. Discussion

The proposed vanishing point estimation method has the following properties: robust, real-time and global optimal.

Robust: The applied 2-line MSS needs only two line segments corresponding to the same vanishing point, which makes the proposed method robust to different scenes with 1, 2 or 3 orthogonal vanishing point(s).

Real-time: The most time-consuming procedure of hypotheses validation is converted into a lookup issue on the polar grid, whose complexity is $\mathcal{O}(n)$, where $n = 37800$ is constant for all the cases. And the time consumption for both polar grid building and hypotheses generation is also very low. Thus the proposed method can achieve a real-time performance of around 40ms on a computer with Intel Core i5-3550p CPU without any optimization and parallel computation. What should be noticed is that, all the three procedures: polar grid building, hypotheses generation and hypotheses validation are parallel inherently, thus they can be accelerated very easily.

Global Optimal: It is very important for a vanishing point estimation algorithm to be global optimal. The RANSAC based methods generally suffer from the inherent problem of the local minima solution. The proposed method, however, can achieve a global optimal solution because the exhaustive strategy is applied. The probability of the 2-line MSS is stable, which is in $[1/3, 1]$, thus the iterative generation of the first vanishing point is reliable. Then the uniform and dense sampling of the second vanishing point with 1° accuracy makes the hypotheses generated adequate to contain at least one hypothesis with 1° close to the ground truth vanishing points. Finally, the exhaustive searching of all the 37800 hypotheses gives a global optimal solution to the vanishing point estimation problem.

3. Experimental Results

We have tested the proposed algorithm on the York Urban Database (YUD) [5]¹ which contains 102 calibrated images with labelled line segments and ground truth vanishing points. Each image contains at least 2 orthogonal vanishing points.

3.1. Measurement

For each ground truth line segment cluster \mathcal{L}_{GT} corresponding to a vanishing point, we traverse each test line segment cluster \mathcal{L}_{test} obtained by a certain algorithm to find the one which satisfies the following conditions:

$$\begin{cases} size(\mathcal{L}_{GT} \cap \mathcal{L}_{test}) > size(\mathcal{L}_{GT})/2, \\ size(\mathcal{L}_{GT} \cap \mathcal{L}_{test}) > size(\mathcal{L}_{test})/2, \end{cases} \quad (8)$$

where $size(\bullet)$ denotes the size of a set. Eq. (8) makes sure that each ground truth line segment cluster can find only one maximal consistent test line segment cluster. The accuracy of the test algorithm is calculated as $\sum_{k=1}^3 N_k / N_{all}$, where N_k stands for the number of the consistent line segments of vanishing point \mathbf{v}_k , and N_{all} is the number of all the line segments.

3.2. Influence of Internal Parameters

The proposed algorithm requires no parameters tuning, since the internal parameters, including the noise ratio (0.5), the confidence coefficient (0.9999), the sampling angle for the polar grid (1°) and the sampling number of \mathbf{v}_2 (360), are all in perfect conditions. The noise ratio is determined by the input line segments and the angle for the polar grid is constant for all the cases, thus in this section, we will show how the confidence coefficient and the sampling number of \mathbf{v}_2 influence the performance of the proposed method.

To test the effect of the confidence coefficient, we set it as 0.9, 0.99, 0.999 and 0.9999, respectively. By setting this, the numbers of iteration are 26, 52, 79 and 105, respectively. Figure 4(a) shows the corresponding accuracies of those different values of the confidence coefficient. We can see that, with the increasing of the confidence coefficient, the accuracy of the vanishing point estimation result is improved step by step. The result of the confidence coefficient as 0.999 (accuracy: 0.992) is approximate to that of the confidence coefficient as 0.9999 (accuracy: 0.993), but the number of iterations of the latter is 26 more than that of the former, which means 9360 more hypotheses in total. However, due to the applied polar grid, the difference of time consumptions between them are small enough to be ignored, thus we still recommend to set the confidence coefficient as 0.9999.

To test the effect of the sampling number of \mathbf{v}_2 , we set it as 60, 120, 180 and 360, respectively, which means the accuracies of the \mathbf{v}_2 are 6° , 3° , 2° and 1° , respectively. Figure 4(b) shows the accuracies of those different values of the sampling number. We can see that, with the increasing of the sampling number, the accuracy also increases despite of the fact that the increment is relatively small. An explanation to this phenomenon is that the distribution of intersection points on the polar grid is sparse, thus even a small sampling number as 60 can achieve the close performance to that of a big sampling number as 360. Still, we recommend a big sampling number as 360 for all the cases.

As a summary, both the confidence coefficient and the sampling number of \mathbf{v}_2 have effect on the final vanishing point estimation accuracy, but the difference is small and the worst accuracy value (0.976) is still very good. Since the time consumption of the hypotheses validation is small, we recommend to set the confidence coefficient as 0.9999 and the sampling number as 360 for all the cases.

3.3. Comparison with Start-of-the-Art Methods

Three start-of-the-art vanishing point estimation methods were tested for comparison with our proposed method, including the Dual Space based method [9]², the 3-line

¹<http://www.elderlab.yorku.ca/YorkUrbanDB/>

²http://dev.ipol.im/~jlezama/vanishing_points/

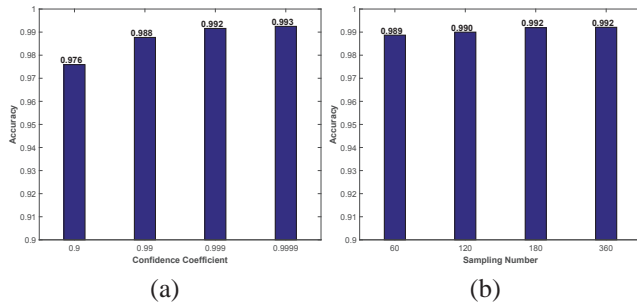


Figure 4. Vanishing point estimation accuracies of difference values of the confidence coefficient (a) and the sampling number (b).

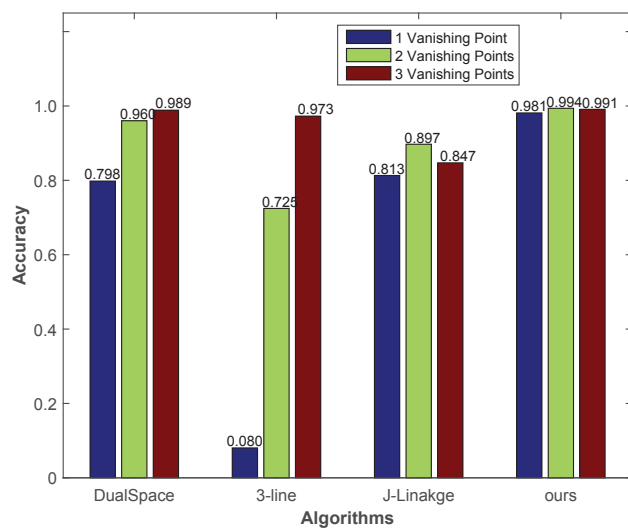


Figure 5. Vanishing point estimation accuracies of the DualSpace based method [9], the 3-line MSS optimal method [11], the J-Linake based method [14] and ours in the YUD, YUD2 and YUD1 datasets.

MSS optimal method [11]³ and the non-iterative J-Linkage based method [14]⁴. The original result of the J-Linkage based method tends to miss some line segments, one reason to this may be that the number of line segments in the YUD is not sufficient enough for the J-Linkage. Thus we re-assigned the line segments to the vanishing points estimated from the original classification result and used the reclassification result as the final output of the J-Linkage based method. We denote these three methods and the proposed algorithm as DualSpace, 3-line, J-Linake and ours, respectively.

Most images of the original YUD contain 3 vanishing points, to test the performance of these methods on the images with 2 and 1 vanishing point(s), we built a 2 vanishing points dataset (denoted as YUD2) and a 1 vanishing point dataset (denoted as YUD1) based on the YUD dataset. The YUD2 dataset is formed by the two vanishing points with

the most and second numbers of corresponding line segments. The YUD1 dataset is formed by the vanishing point with the most number of corresponding line segments. Figure 5 shows the vanishing point estimation accuracies of these four methods in the YUD, YUD2 and YUD1 datasets, from which we can get the following observations. First, in all the datasets, the proposed method achieves the best performance among these four methods, especially in the YUD2 and YUD1 datasets which contain 2 and 1 vanishing point(s), respectively. Second, the DualSpace based method and the J-Linake based method are both useful on the image with 2 or 1 vanishing point(s), but their accuracies are much lower than that of our method. On the YUD1 dataset with only one vanishing point, the accuracies of the DualSpace based method and the J-Linake based method are 0.798 and 0.813, respectively, while that of our method is 0.981. Third, the 3-line MSS optimal method failed in the YUD1 dataset with an accuracy of 0.080, because the 3-line MSS is based on the assumption that there are at least 2 vanishing points on the image. Fourth, the accuracies of the J-Linake based method are moderate in all the YUD, YUD2 and YUD1 datasets, this is for the reason that the J-Linake based method was designed to detect the line segment clusters with the same pattern, thus the J-Linake based method can be applied in the cases of 3, 2, or 1 vanishing point(s). However, due to the lack of the orthogonal constrain, the J-Linkage based method can not achieve very high accuracies. Fifth, even in the original YUD dataset with 3 vanishing points, the proposed method still achieve the highest accuracy 0.991. As a summary, the proposed method is robust on an image with 3, 2 or 1 vanishing point(s), and can achieve a higher vanishing point estimation accuracy than the start-of-the-art methods. Figure 6 is a demonstration of the vanishing point estimation results of the proposed method on two representative images of the YUD dataset, from the left to the right are the original line segments, the results of 3 vanishing points, 2 vanishing points and 1 vanishing point, respectively.

3.4. Robustness to Outliers

To validate the robustness of the proposed method to outliers, we tested it on the YUD dataset with different ratios of outliers. For each image in the YUD dataset, we randomly selected 10%, 20%, 30%, 40%, 50%, 60% and 70% of its line segments as outliers by adding noises on the orientation of these line segments. For a line segment, the orientation noise was added by rotating it round its midpoint with a random angle in $[10^\circ, 20^\circ]$. Then, those line segments were considered as outliers since the angle deviation between it and the corresponding vanishing point was larger than 10° . We tested the proposed method, the DualSpace based method, the 3-line MSS optimal method and the J-Linake based method in these YUD dataset with dif-

³<http://www-users.cs.umn.edu/~faraz/?p=research>

⁴<http://www-etud.iro.umontreal.ca/~tardifj/>

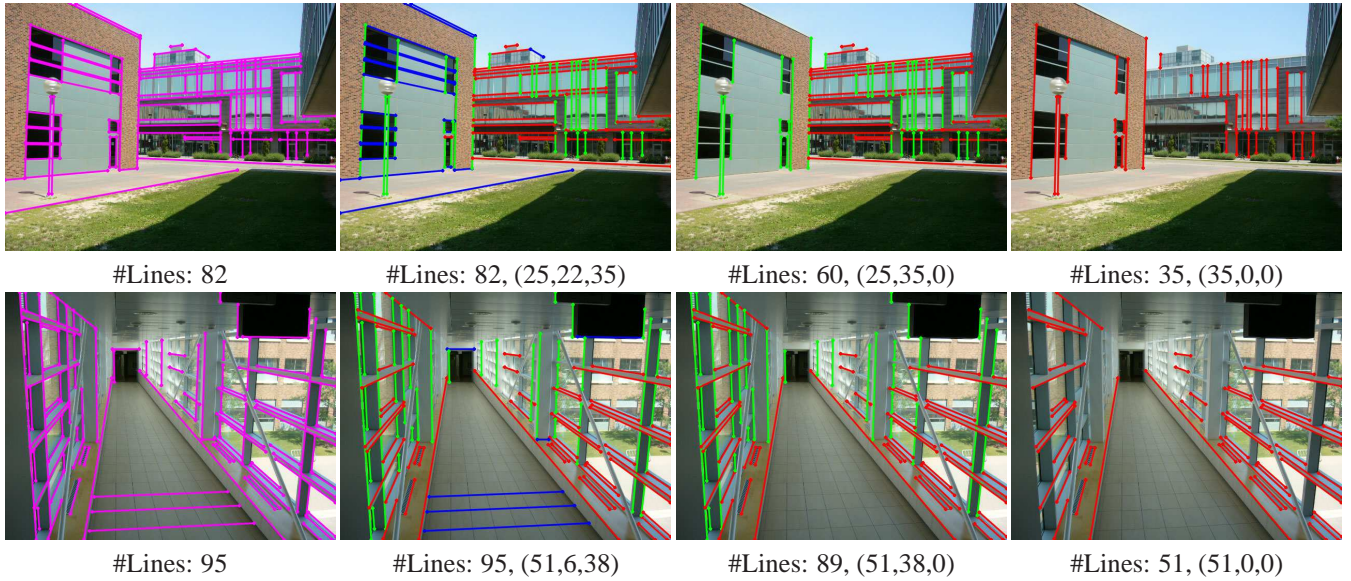


Figure 6. Vanishing point estimation results of the proposed method on the YUD dataset with 3, 2, and 1 vanishing point(s). The triple “(, ,)” denotes the numbers of line segments corresponding to different vanishing points, respectively.

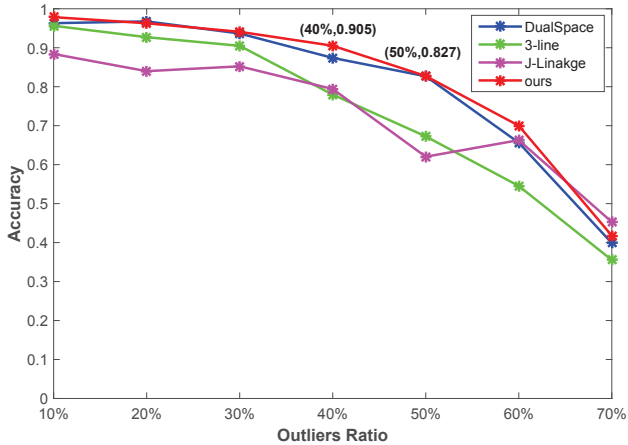


Figure 7. Vanishing point estimation accuracies of the DualSpace based method [9], the 3-line MSS optimal method [11], the J-Linake based method [14] and ours in the YUD dataset with different outliers ratios.

ferent outliers ratios respectively. Figure 7 shows the vanishing point estimation accuracies of these four methods in the YUD dataset with different outliers ratios, from which we can draw the following conclusions. First, the accuracies of all these methods drop with the increasing of outliers ratio. Second, the proposed method can achieve the highest accuracies among all these four methods. Third, the accuracy of our method is higher than 0.9 when the outliers ratio is 40%, and is 0.827 even when the outliers ratio is 50%, which demonstrates the robustness of the proposed method. Fourth, our method can't perform well when the outliers ratio is larger than 50%.

3.5. Performance on Real Scene Images

The line segments of the YUD dataset contains no outliers, which is an ideal case. In practice, the vanishing points are estimated based on the line segments obtained by a certain line segment detector, which often contains many outliers. To evaluate the robustness of the proposed method on the real scene images, we tested it on the line segments detected by LSD [17], and discarded the line segments shorter than 30 pixels. Figure 8 shows the line segments detected by LSD (the first and third columns), and the line segment clusters estimated by our method (the second and fourth columns). We can see in Figure 8 that the proposed method performed very well in the real scene images, the three orthogonal vanishing points were estimated and the outliers were discarded well. What should be noticed is that the time-consumption of the proposed method on these images is around 40ms on average without any optimization and parallel computation on a computer with Intel Core i5-3550p CPU, which shows the ability of our method to be applied on real-time and real-scene applications like indoor navigation and so on.

4. Conclusion

This paper presents a very simple and efficient algorithm to estimate 1, 2 or 3 orthogonal vanishing point(s) on a calibrated image in Manhattan world. To generate the vanishing point hypotheses, we propose to use 2 lines to get the first vanishing point v_1 , then uniformly take sample of the second vanishing point v_2 , and finally calculate the third vanishing point v_3 via the cross-product of v_1 and v_2 . To obtain a global optimal solution to the vanishing point estima-



Figure 8. Vanishing point estimation results of the proposed method in the YUD dataset with line segments detected by the LSD [17]. The line segments detected by LSD are drawn in pink in the first and third columns, the line segments corresponding to different vanishing points are drawn in different colors and the outliers are marked in black in the second and fourth columns.

tion problem, we applied an exhaustive searching strategy which validates every hypothesis to find out the best one. To accelerate the time-consuming exhaustive searching procedure, a polar grid is built, which converts this problem into a lookup issue whose complexity is $\mathcal{O}(n)$. To demonstrate the performance of the proposed method, we compared it with three start-of-the-art methods in the YUD dataset and its variants. Experimental results show that our method out-

performed the start-of-the-art methods, especially on the images with 2 and 1 vanishing point(s). We also experimented on the line segments detected by the LSD detector, it turns out that our method can still achieve very good performance.

References

- [1] M. E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 282–289, 2000.
- [2] J. C. Bazin, C. Demonceaux, dric, P. Vasseur, and I. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. *International Journal of Robotics Research*, 31(1):63–81, 2012.
- [3] J. C. Bazin and M. Pollefeys. 3-line RANSAC for orthogonal vanishing point detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4282–4287, 2012.
- [4] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, 1990.
- [5] P. Denis, J. H. Elder, and F. J. Estrada. Efficient edge-based methods for estimating Manhattan frames in urban imagery. In *European Conference on Computer Vision*, pages 197–210, 2008.
- [6] D. H. Hedau, Varsha and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *IEEE International Conference on Computer Vision*, pages 1849–1856, 2009.
- [7] K. Ikeuchi, P. Vasseur, C. Demonceaux, I. Kweon, Y. Seo, J. C. Bazin, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645, 2012.
- [8] J. K. Lee and K. J. Yoon. Real-time joint estimation of camera orientation and vanishing points. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1866–1874, 2015.
- [9] J. Lezama, R. G. V. Gioi, G. Randall, and J. M. Morel. Finding vanishing points via point alignments in image primal and dual domains. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 509–515, 2014.
- [10] M. J. Magee and J. K. Aggarwal. Determining vanishing points from perspective images. *Computer Vision Graphics & Image Processing*, 26(2):256–267, 1984.
- [11] F. M. Mirzaei and S. I. Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *IEEE International Conference on Computer Vision*, pages 2454–2461, 2011.
- [12] C. Rother. *Multi-View Reconstruction and Camera Recovery Using a Real or Virtual Reference Plane*. PhD thesis, Royal Institute of Technology, 2002.
- [13] C. Rother. A new approach to vanishing point detection in architectural environments. *Image & Vision Computing*, 20(910):647–655, 2002.
- [14] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *IEEE International Conference on Computer Vision*, pages 1250–1257, 2009.
- [15] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. In *European Conference on Computer Vision*, pages 537–547, 2008.
- [16] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *International Journal of Computer Vision*, 97(3):305–321, 2012.
- [17] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.
- [18] H. Wildenauer and A. Hanbury. Robust camera self-calibration from monocular images of manhattan worlds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2831–2838, 2012.
- [19] L. Zhang, H. Lu, X. Hu, and R. Koch. Vanishing point estimation and line classification in a manhattan world with a unifying camera model. *International Journal of Computer Vision*, 117(2):113–130, 2016.