

A Method for Estimating the Camera Parameters Based on Vanishing Points

Wan Fang, Li HaiNing, Jin HuaZhong^(✉), Lei GuangBo,
and Ruan Ou

Hubei University of Technology, Wuhan, China
devwaf@qq.com, galaxy0522@163.com

Abstract. This paper presented MMVD vanishing points detection algorithm which was inspired by multi-model estimation. It was operated in image pixel space, not parameter space. We first gave some vanishing points estimation model assumption based on the lines' position and angle of each other. Then, the model assumption was used to construct the correlation matrix for line segments clustering and vanishing points estimation. Clustering algorithm produced vanishing points and they were refined by the EM algorithm to improve the accuracy. Experiments showed that the detection speed was good and the detection result can ensure good accuracy.

Keywords: Multi-model · EM · Clustering · Vanishing point

1 Introduction

In graphical perspective, a vanishing point was a point in the picture plane that was the intersection of the projections (or drawings) of a set of parallel lines in space on to the picture plane. When the set of parallels was perpendicular to the picture plane, the construction was known as one-point perspective and their vanishing point corresponds to the oculus or eye point from which the image should be viewed for correct perspective geometry.

Based on the vanishing point, we could calibrate the camera. But there were two key problems. One was the detection of vanishing point. The precision of vanishing point directly determined the accuracy of camera calibration. If the point accuracy was not high, it could lead to camera calibration error. On the other hand, computation efficiency was crucial. Only the fast vanishing point detection method could be used for real-time large-scale 3d reconstruction, but most of the existing vanishing point detection algorithms were unable to meet the requirements. Bazin [1] said, in the past 30 years, although a lot of research was carried out, but no one was completely satisfactory, the difficulty was that this was a pathological problem. For example, the iterative detection algorithm of vanishing point often produced local minima, the orthogonality calculation of vanishing point was complex, and the data may contain a large number of outliers and so on. Even so, because the camera self-calibration could provide an automatic way to obtain the initial position and orientation of the camera which was valuable for image-based 3d reconstruction, the rapid detection of vanishing point had been highly valued by researchers.

2 Related Research

Caprile was first to use vanishing point to calculate the main point and rotation matrix [2], many researchers carried out related work, but they all assumed that part of camera parameters were known. Haralick [3] did not directly use vanishing point, using a rectangular projection in the image. Wilczkowiak [4] used the relationship between parallel lines to estimate projection matrix, whose essence was using of vanishing point calibration. Chen [5] used a uniform grid on the assumption that the focal length was known, which was the equivalent of artificial designated two vanishing point direction. These methods have the requirements of constraint conditions, which are specified in a single image by hand.

Recent year, Orghidan [6], Lopes [7] and Dong [8] used structured light features to calibrate the camera, indirectly used vanishing points. Wang [9] used a hexahedral as the calibration target to generate vanishing line. These vanishing lines contained effective geometric constraint information for the calculation of camera direction and focal length, but this calibration condition was too strict. Grammatikopoulos [10] used multiple images, each of which was calibrated by the internal reference of the camera. Using the direct geometric reasoning method, the absolute quadric curve was used in essence. Because only two vanishing points existed on multiple images, his algorithm could be addressed to the polygon like objects. Similarly, in the method of Stevenson [11], the calibration equation was simplified by motion control of multiple cameras.

These calibration methods basically needed constraint conditions such as camera moving track, or some manually specified marks in the image. In fact, the geometrical characteristic of structured 3d scene in images itself could provide similar constraints for camera calibration.

3 MMVD Algorithm

LSD [12] proposed a method of line segment detection which could provide linear time complexity, and the error rate was very low. We took it to help detect line segments.

In this paper, according to the features of 3D scene structure, we put forward a MMVD vanishing point detection algorithm for multi-model estimation. Multi-model estimation method was inspired by Toldo [13]. The workflow of the algorithm was as described:

- (1) Construction of line subset and vanishing point model hypothesis
- (2) Compute a Correlation matrix
- (3) Line segments clustering and vanishing point estimation
- (4) EM optimization

The following sections will focus on the process of each step to describe the algorithm in detail. Here it was some definitions used in the algorithm:

L denoted an array of line segments detected by line segments, L_s denoted a line subset of L , and was a collection of line segments used for vanishing point estimation. e_n signified the detected n segment, e_n^s denoted the starting point coordinate of line

segment n , e_n^e denoted the end coordinates of the line segments n , and e_n^m denoted the midpoint coordinates of line segment n .

3.1 Construction of Line Subset and Vanishing Point Model Hypothesis

Model hypothesis was only the initial value which would be modified in repeated calculation. Original line segments set we named it L , which usually contained a lot of external data or error data, so line subset was to be constructed to remove these data points. Vanishing point model hypothesis was also built from this subset.

SampleSetConstruct algorithm was described as followed:

```

Input: L parameters: M (vanishing point count)
Output: Ls, lineSeg[] and VP[] (M) initial vanishing point
estimation
1  foreach line l in LineSegAll[] do
2      for(i=0;i++;i<M )
3          if lineSeg[i].size<size then
4              addLineSeg(lineSeg[i]); //find the nearest line
5          else
6              ModelHypothesis(lineSeg[i]) //generate VP
7              endif
8      endfor
9  endfor

```

Size of L_s (line segments) was N . Assuming that there were M vanishing points estimations, then the algorithm traversed in L space. Each two line segments formed a minimum sampling set $lineSeg[i]$, and estimated a vanishing point. Model Hypothesis calculated intersection of line pair, as the model hypothesis of vanishing point. We could think that the hypothesis was not accurate, but it had a theoretical basis to be an initial value, which was the spatial relationship of line segment and vanishing point. Sample Set Construct could find pair for each line segment, unless it was obviously an outlier. Figure 1 described the process of constructing the model hypothesis and vanishing point line subset.

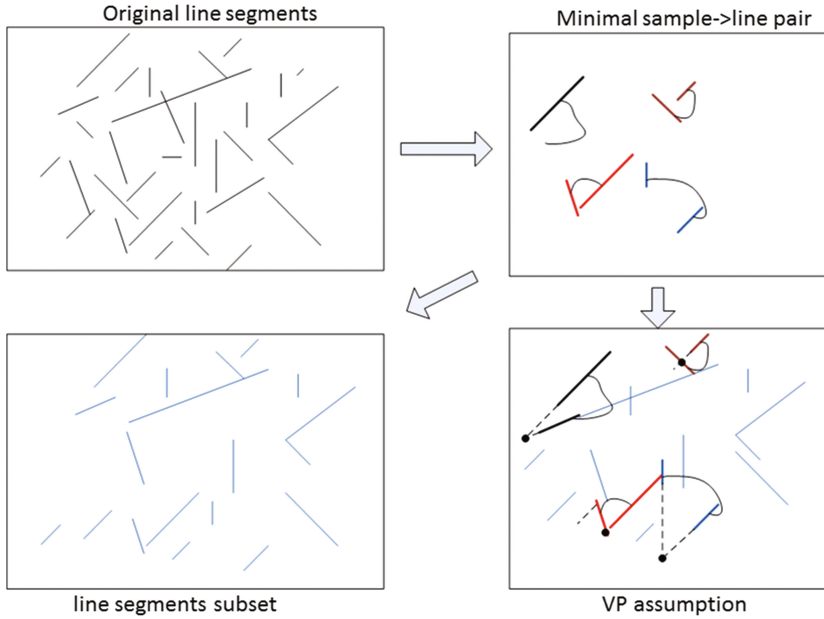


Fig. 1. From the original line segments, we can easily classify them into one-one pair form, so, each pair had two lines and became the start of clustering. From now on, line subset and model hypothesis of vanishing point was constructed iteratively.

3.2 Correlation Matrix

Correlation matrix determined the correlation of L_s and M vanishing points, which constructed a consistency metric relation. Set center line of v_m and e_n as l_n , we should compute the angle between the reference line l_n and line segment e_n . If θ was greater than a certain threshold, we asserted that there was no correlation between line and vanishing point (Fig. 2).

$$\theta(e_n, l_n) = \arccos \frac{e_n \cdot l_n}{|e_n||l_n|} \tag{1}$$

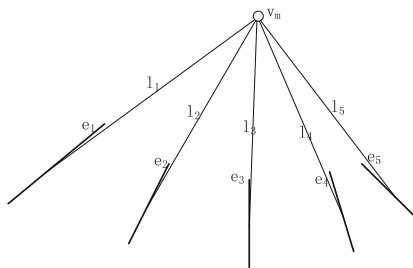


Fig. 2. We used reference line to estimate relationship between v_m and e_n

In the correlation matrix $R_{N \times M}$, each row were N segments of the subset L_s , column represented M models of vanishing point. Each element of matrix was valued as the correlation degree:

$$R_{N \times M} = R(v_m, e_n) \quad (2)$$

When the correlation matrix built, hypothesis of position of vanishing point obtained support on numerical value and relationship between vanishing point and line segment formed a good initial state of clustering iteration. Next was the clustering and estimation.

3.3 Line Segments Clustering and Vanishing Point Estimation

When the initial state of clustering was ready, in this step, the main work was to cluster the line in `RelateMatrix`, and generate vanishing point estimation for each cluster. Obviously, this was an iteration process. Vanishing point hypothesis in previous section could be modified. Clustering method decided which line belong to which class and also decided the result of vanishing point. In the clustering process, we used Jaccard [14] distance to classify the data. At the end of the iteration, the distance between all line segments is less than the specified threshold D .

Then it was to estimate vanishing point of each cluster. Line segment clustering of each step resulted in the estimation value of a multi-model. Each model corresponded to a vanishing point direction. Calculation formula according to the distance of the vanishing point was as followed:

$$v_j = \operatorname{argmin} \sum_{i,j=(1..m)} d(e_i, vp_j) \quad (3)$$

The formula meant that for the cluster, the point could be the vanishing point because it had the minimum average distance to lines in this cluster. This formula was good at computation and weak at accuracy, which avoided least square solution.

In the process of clustering, each class was taken into calculation several times until class distance was greater than a threshold D . The number of iterations depended on the distribution of line segments in the image and the setting of the distance threshold.

3.4 EM Optimization

In this step, we tried to overcome our weakness in vanishing point estimation accuracy. The clustering process described above most likely to split multiple classes which corresponded to same vanishing point. EM method [15] was to refine the clustering results for the purpose of optimization. Now, we wanted to obtain vanishing point estimation of high reliability with EM optimization. In this section, we designed an EM algorithm, while the n lines and m vanishing points were known. It was transformed into a problem of using maximum likelihood function to estimate direction of lines.

$$\max \prod_{i=1}^M p(v_m) = \sum_{i=1}^M \log p(v_m) \tag{4}$$

assumed:

$$p(v_m|e_n) = \frac{1}{w_{mn}\sqrt{2\pi}\sigma_i} \exp\left(\frac{-\theta_{mn}^2}{2\sigma_i^2}\right) \tag{5}$$

Among them, $\theta_{mn} = \sin^{-1}(v_m \cdot e_n)$, was angle of line direction and the direction of the vanishing point. w_{mn} was the weight, which could be obtained by calculating the largest eigenvalue of the symmetric matrix, where C_m was the covariance matrix of the edge.

In M step, σ_i^2 , $p(e_n)$ and e_n was estimated to make a maximum likelihood function.

$$p(e_n) \approx \frac{1}{N} \sum_{m=1}^M p(e_n|v_m) \tag{6}$$

$$\sigma_i^2 \approx \frac{\sum_{m=1}^M \theta_{mn}^2 p(e_n|v_m)}{Np(e_n)} \tag{7}$$

Because $\theta_{mn} = \sin^{-1}(v_m \cdot e_n) \approx v_m \cdot e_n$, then e_n can be used with the right direction of the linear least squares estimation formula:

$$\min_{e_n} \left\| (W_n^{m \times m} A_n^{m \times 3} e_n) \right\|^2 \tag{8}$$

W was a diagonal matrix containing weight $p(e_n|v_m)$, A was vanishing point matrix. Through the SVD decomposition, we could to find the vector that was related to the direction of line e_n .

4 Experiment

4.1 Efficiency of Vanishing Point Estimation

We took York Urban database [16] as our experiment data, including indoor and outdoor images. Camera model was Panasonic DMC-LC80, image size at 640 * 480.

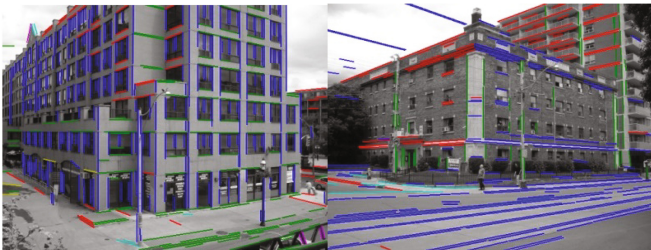


Fig. 3. Lines extracted in orthogonal directions of outdoor scenes



Fig. 4. Lines extracted in orthogonal directions of indoor scenes

The following figure was the result of line detection and the three vertical edges were obtained by line segment clustering (Figs. 3 and 4).

We compared our method with Denis’ [16]. Table 1 showed the performance results of vanishing point estimation. In the step of line segment extracting, several algorithms were quite like in performance. But in the step of vanishing point estimation, MMVD algorithm and MMVD + EM had obvious advantages.

Table 1. VP estimation performance comparison

| Algorithm | Lines number | Line extracting time(s) | VP estimation time (s) | Total Time(s) |
|-----------|--------------|-------------------------|------------------------|---------------|
| Denis | 1532 | 67 | 1445 | 1512 |
| MMVD | 1021 | 45 | 513 | 558 |
| MMVD + EM | 1021 | 45 | 535 | 581 |

It could be seen that for images of different sizes, even using EM optimization, the convergence speed was very fast, it only took a few iterations to end. For images with size of 1024 * 1024 or larger, Denis algorithm failed.

The images in Figs. 5 and 6 were rendered from the scene of a real-time digital city with a resolution of 2048 × 1024. For the large image, the MMVD algorithm could perform within 20 s, which could meet the needs of practical application.

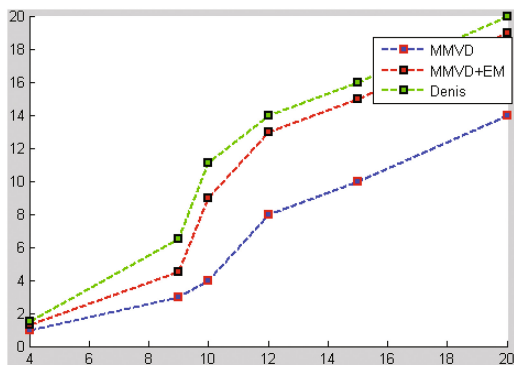


Fig. 5. Performance comparison of several algorithms on different size images

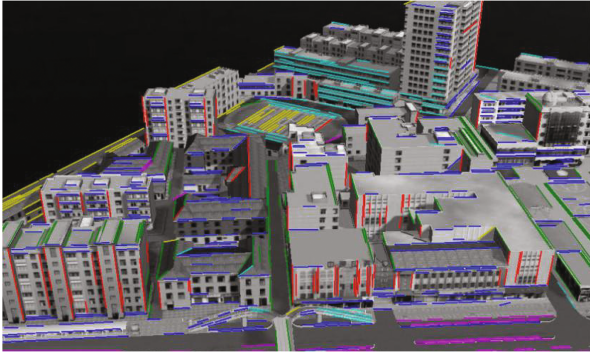


Fig. 6. Line extraction results of scene rendering in digital city project

4.2 Accuracy of Vanishing Point Estimation

For the vanishing points detected by MMVD algorithm, we used the algorithm proposed by Caprile [2] to calculate the focal length. And, we compared to the focal length provided by York Urban database to verify the accuracy of our estimation algorithm (Table 2).

Table 2. Comparison of principal distance and real principal

| Image Num | P1020826 | P1020830 | P1020833 | P1020177 | P1020847 |
|----------------|----------|----------|----------|----------|----------|
| MMVD Focal | 672.5933 | 672.5621 | 671.55 | 672.3515 | 672.2451 |
| Original Focal | 672.6122 | 672.6122 | 672.6122 | 672.6122 | 672.6122 |

The experiment used indoor and outdoor images. We used algorithm proposed by Caprile to calculate the focal length. From the experimental results, it could be seen that the error of camera focal length and the actual focal length calculated is small, proving high accuracy of the estimate algorithm.

5 Summary

An important feature of three-dimensional structural scene was the presence of three orthogonal directions, each orthogonal direction in the finite or infinity image plane formed vanishing point. The orthogonal vanishing points can be used for the self-calibration of the camera, and was more convenient than other manual calibration methods. In this paper, a vanishing point detection method of multi-model consensus estimation was proposed. This method used the detected line, using line segment clustering to achieve the estimation of multiple vanishing point, and then optimized the results through the EM algorithm. Experiments proved that, the algorithm could quickly and accurately estimate the vanishing point and compute camera parameters.

Acknowledgement. The work was supported by the Educational Commission of Hubei Province of China (No. D20151401) and the Green Industry Technology Leading Project of Hubei University of Technology (No. ZZTS2017006).

References

1. Bazin, J.C., Seo, Y., Démonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., Pollefeys, M.: Globally optimal line clustering and vanishing point estimation in Manhattan world. In: CVPR, pp. 638–645 (2012)
2. Caprile, B., Torre, V.: Using vanishing points for camera calibration. In: IJCV, pp. 127–40 (1990)
3. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision, vol. 2. Addison-Wesley, Boston (1993)
4. Wilczkowiak, M., Boyer, E., Sturm, P.: Camera calibration and 3D reconstruction from single images using parallelepipeds. In: Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 142–148. IEEE (2001)
5. Chen, W., Jiang, B.C.: 3-D camera calibration using vanishing point concept. *Comput. Vis. Pattern Recogn.* **24**, 57–67 (1991)
6. Orghidan, R., Salvi, J., Gordan, M., et al.: Structured light self-calibration with vanishing points. *Mach. Vis. Appl.* **25**(2), 489–500 (2014)
7. Lopes, F., Silva, H., Almeida, J.M., et al.: Structured light system calibration for perception in underwater tanks. In: Iberian Conference on Pattern Recognition and Image Analysis, pp. 111–120 (2015)
8. Dong, S., He, B., Lin, C., et al.: Calibration method for a structured light measurement system with two different focal length cameras. *Measurement* **73**, 462–472 (2015)
9. Wang, L.L., Tsai, W.H.: Camera calibration by vanishing lines for 3-D computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 370–376 (1991)
10. Grammatikopoulos, L., Karras, G., Petsa, E.: Camera calibration combining images with two vanishing points. *Int. Arch. Photogrammetry Remote Sens. Spat. Inf. Sci.* **35**(5), 99–104 (2004)
11. Stevenson, D., Fleck, M.: Robot aerobics: four easy steps to a more flexible calibration. In: ICCV 1995, p. 34–39 (1995)
12. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., et al.: LSD: a fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(4), 722–732 (2010)
13. Toldo, R., Fusiello, A.: Robust multiple structures estimation with J-Linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 537–547. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88682-2_41](https://doi.org/10.1007/978-3-540-88682-2_41)
14. Jaccard, J. (ed.): *Interaction Effects in Logistic Regression*, vol. 135. Sage Publications, Thousand Oaks (2001)
15. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. Ser. B (Methodological)*, pp. 1–38 (1977)
16. Denis, P., Elder, J., Estrada, F.: Efficient edge-based methods for estimating manhattan frames in urban imagery. In: European Conference on Computer Vision, part II, pp. 197–210 (2008)