The 3-Point Method: A Fast, Accurate and Robust Solution to Vanishing Point Estimation

Vinod Saini, Shripad Gade, Mritunjay Prasad, Saurabh Chatterjee
Department of Aerospace Engineering,
Indian Institute of Technology, Bombay,
Mumbai, India-400076
{saini.vinod, shripad.gade, mritunjay, saurabhsaurc}@iitb.ac.in

ABSTRACT

Vanishing points can provide information about the 3D world and hence are of great interest for machine vision applications. In this paper, we present a single point perspectivity based method for robust and accurate estimation of Vanishing Points (VPs). It utilizes location of 3 collinear points in image space and their distance ratio in the world frame for VP estimation. We present an algebraic derivation for the proposed 3-Point (3-P) method. It provides us a non-iterative, closed-form solution. The 3-P results are compared with ground truth of VP and it is shown to be accurate. Its robustness to point selection and image noise is proved through extensive simulations. Computational time requirement for 3-P method is shown to be much less than least squares based method. The 3-P method is extremely useful for accurate VP estimation in structured and well-defined environments.

Keywords

Vanishing Points, Point Perspectivity, Length Ratio, Camera Calibration, Cross Ratio

1 INTRODUCTION

A family of parallel lines projected on a plane under the pin-hole camera model will ideally intersect in a common point. This point is known as the Vanishing Point. VP's formed by families of coplanar parallel lines are collinear and the line is known as the Vanishing Line. VPs and Vanishing Lines for an image of a cube are shown in Fig. 1.

Development of computational techniques and ever-growing requirement of extracting information from image have led to a spurt in the field of image analysis in recent years. Vanishing points have myriad applications including camera calibration, robotic navigation, 3D reconstruction, pose estimation, augmented reality etc. VP's have been extensively used for camera calibration. [7], [9], [10], [11] and [12] use VPs for camera calibration. Image reconstruction probelm in [13] and [14] use VP's for extraction of 3D coordinates of points. [15] uses parallel lines in the environment and corresponding VP's for steering a robot. [16] uses vanishing points and vanishing lines for pose estimation of UAV's in indoor flights.

Several methods have been proposed to estimate VP's. [3] uses J linkage based algorithm for vanishing point estimation in man-made environments. [4] proposes a new framework for line based geometric analysis and VP estimation of Manhattan scenes. [5] uses accurately localised edges that are obtained through edge pixels and does not require fitting of lines. It uses fewer but more accurate lines for estimating VPs. [8] relies line extraction using Hough transform and then on voting in the vanishing point space to estimate VP. All of the above mentioned methods claim to be accurate for VP estimation in architectural environments e.g. Manhattan scenes.

Vanishing points are extremely important in computer vision and the accuracy of VP estimation directly influences the performance of the said application. Additionally, since there may be a need to compute vanishing points multiple times, a computationally efficient method is the need of the day. Camera calibration efforts require accurate VP estimates Calibration targets are well defined structured objects and this information about them can be exploited to suit our needs. Fig. 2 shows few calibration patterns used in different algorithms. [1] proposes two length ratio based methods. First requires evaluation of 1D projective transformation and the direction of lines to compute VP. Second detects VP using geometric construction. Here we propose a length-ratio based fast and accurate VP estimation method. We use three collinear points and their distance ratio in world frame to compute VP location.

 ISBN 978-80-86943-75-6

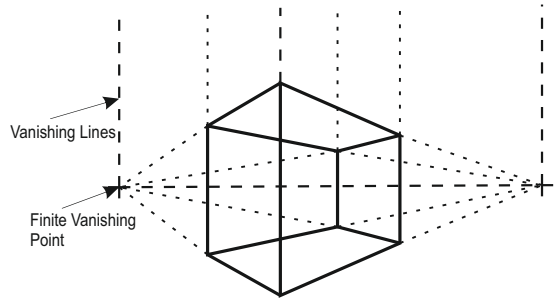

Figure 1: Vanishing Points and Vanishing Line

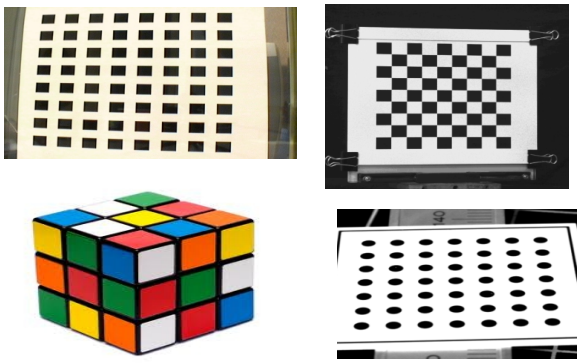

Figure 2: Patterns used to get VPs

Section 2 discusses camera calibration preliminaries and least squares vanishing point detection method. Section 3 talks about the 3-P method and its formal derivation. Simulation and experimental results are discussed in section 4.

2 PRELIMINARIES

2.1 Camera Model

Pin-hole model is based on the principle of collinearity, where each point in the world space can be mapped by a straight line to the image plane through the camera center. This kind of central projective transform is called “Perspective". Fig. 3 shows projection of a line on image plane. A Pin-hole camera model has been used here ([1], [2]). Any point P (coordinates given by **X**) can be mapped to a point p (coordinates given by **x**) in the image plane . The overall transform can be expressed as,

$$\mathbf{x} = \mathbb{P}\mathbf{X} \tag{1}$$

The overall transformation matrix, $\mathbb{P}$ is obtained by multiplying the extrinsic calibration matrix by intrinsic calibration matrix. $\mathbb{P}$ can be expressed as,

$$\mathbf{x} = \mathbb{K}\,R[I \mid -\tilde{C}]\mathbf{X} = \mathbb{K}\,\mathbb{M}\,\mathbf{X} = \mathbb{P}\,\mathbf{X} \tag{2}$$

Parameters that are solely dependent on camera are called Intrinsic parameters. Principal point, skew, aspect ratio and focal length together form the intrinsic camera calibration matrix ($\mathbb{K}$). Extrinsic parameters of a camera include the rotation and translation of camera with respect to the world frame. Rotation matrix ($\mathbb{R}$) and translation vectors ($\tilde{C}$) together form the extrinsic camera calibration matrix ($\mathbb{M}$).

2.2 Least Squares VP estimation

Let us consider a set of n parallel lines in 3D. Ideally their mappings in the image plane will intersect at a point (the VP). Due to noise and other errors they will intersect at different points. The maximum number of intersections that can be found out are ${}^nC_2$. The aim is to estimate a point that has least perpendicular distance from the n lines. The methodology can be divided into two sub-tasks,

1. Extracting Lines from the image.
2. Finding Least Squares solution.

Extracting Lines Lines can be extracted from the image using various image processing techniques. In our approach we extract control point locations from the image (see Figure 8). A least squares fit straight line is drawn to minimize perpendicular distance of m points from the line. If (x_i, y_i) are the locations of the m points that lie on a straight line, the slope (slope) and intercept (c) of the line are given by,

$$\begin{bmatrix} c \\ slope \end{bmatrix} = \begin{bmatrix} m & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i y_i \end{bmatrix} \tag{3}$$

Least Squares solution Once the line information is extracted from the image, the only hurdle in estimating the vanishing point using least-squares is finding the intersection of lines. For any two points v_i and v_j, the line passing through them can be expressed as,

$$L_{ij} = v_i \times v_j$$

If m lines given by $L_{i\,(i=1,2,\ldots,m)}$ intersect in a point V, then the coordinates of V are given by,

$$\begin{bmatrix} L_1^T \\ .. \\ L_m^T \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = 0 \tag{4}$$

3 THE 3-POINT METHOD

Parallel lines appear to intersect at a point in perspective view. In an image this perspectivity is introduced due the camera parameters and its orientation. Several methods have been proposed in literature to measure perspectivity. In [1] this perspectivity is measured

 ISBN 978-80-86943-75-6

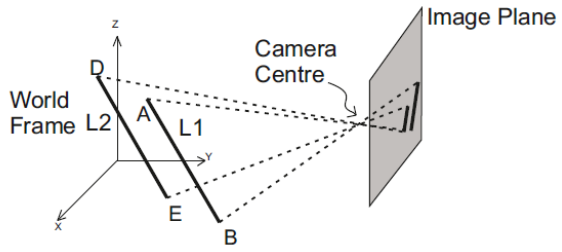

Figure 3: Pin-hole camera model

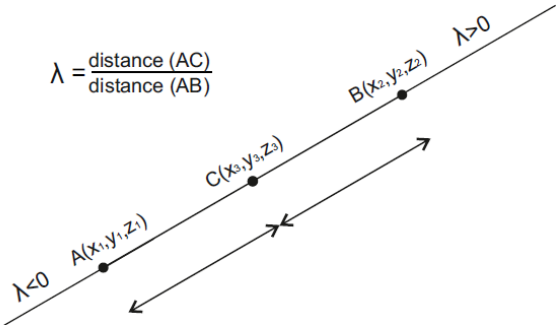

Figure 4: Length Ratio for 3-P method

through the evaluation of 1D projective transformation. The main idea behind or method is that we can get a sense of this camera perspectivity through the three collinear points from the image and their length ratio in the world frame.

Let us consider a family of parallel lines represented by $\mathscr{F}$. These lines are projected onto an image plane through a pin-hole camera model. Let this set be called as $\mathbb{F}$. Now, if we select two points lying on any line f (f $\in$ $\mathbb{F}$) ; we can write the equation of that line f in image space. Equation of line f and the length ratio (given by three collinear points) in the world frame will provide information about perspectivity along f. This will enable us to map any point on line f ($f \in \mathscr{F}$) onto its image f (f $\in \mathbb{F}$). Any point on line f at infinite distance when mapped under pin-hole camera model will map onto VP.

The advantage of this method over the length ratio method is that it does not require us to compute homography (projective transform) and perform intensive computations. It provides us with a closed-form solution and is computationally efficient. Line detection and clustering is not required in this method. This reduces computational load significantly and altogether eliminates line detection and clustering errors. Also this method can be made to utilize information from a small region in the image, thereby reducing errors due to defocusing of certain parts of image.

3.1 Derivation

Let us consider three collinear points A(x_1,y_1,z_1), B(x_2,y_2,z_2) and C(x_3,y_3,z_3). Let, the line that passes through them be called L_1 as shown in Fig. 3.
Distance Ratio: The distance ratio for three collinear points is given by (see Fig. 4),

$$\Gamma = \frac{d_{AC}}{d_{AB}} = \frac{\sqrt{(x_1-x_3)^2+(y_1-y_3)^2+(z_1-z_3)^2}}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}} \quad (5)$$

Camera Model: We use a pin-hole camera model with projection matrix $\mathbb{P}$. Let the images of A, B and C be called A'$(u_1,v_1,1)$, B'$(u_2,v_2,1)$ and C'$(u_3,v_3,1)$ respectively.

$$A' = \mathbb{P}A,\ B' = \mathbb{P}B,\ \text{and}\ C' = \mathbb{P}C \quad (6)$$

Line L_1: Equation of line L_1 (see Fig. 3) can be written in the two point form (using points A and B) as follows,

$$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} = \frac{z-z_1}{z_2-z_1} = \lambda \quad (7)$$

or

$$\begin{aligned} x &= x_1 + \lambda(x_2-x_1) \\ y &= y_1 + \lambda(y_2-y_1) \\ z &= z_1 + \lambda(z_2-z_1) \end{aligned} \quad (8)$$

Now, if we substitute coordinates of point C in Eq. (8) and use the expression in Eq. (5) we can easily conclude that Γ and λ are equal.
Coordinates of C', for a known distance ratio, can be expressed as,

$$\begin{aligned} \begin{pmatrix} w_3u_3 \\ w_3v_3 \\ w_3 \end{pmatrix} &= \mathbb{P}\begin{pmatrix} x_1+\lambda(x_2-x_1) \\ y_1+\lambda(y_2-y_1) \\ z_1+\lambda(z_2-z_1) \\ 1 \end{pmatrix} \\ &= \mathbb{P}\begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} + \lambda\mathbb{P}\begin{pmatrix} x_2-x_1 \\ y_2-y_1 \\ z_2-z_1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} w_1u_1 \\ w_1v_1 \\ w_1 \end{pmatrix} + \lambda\begin{pmatrix} w_2u_2-w_1u_1 \\ w_2v_2-w_1v_1 \\ w_2-w_1 \end{pmatrix} \end{aligned} \quad (9)$$

Simplifying the above equation we get C' as,

$$(u_3,v_3) = \left(\frac{u_1+\lambda(\alpha u_2-u_1)}{1+\lambda(\alpha-1)}, \frac{v_1+\lambda(\alpha v_2-v_1)}{1+\lambda(\alpha-1)} \right) \quad (10)$$

where, α is defined as $\frac{w_2}{w_1}$.

For any point D(x,y,z) lying on line L1, and the image D'$(u,v,1)$ are related by $D' = \mathbb{P}D$. The coordinates of D' are obtained by Eq. 10, and expressed as,

$$(u,v) = \left(\frac{u_1+\lambda'(\alpha u_2-u_1)}{1+\lambda'(\alpha-1)}, \frac{v_1+\lambda'(\alpha v_2-v_1)}{1+\lambda'(\alpha-1)} \right) \quad (11)$$

 ISBN 978-80-86943-75-6

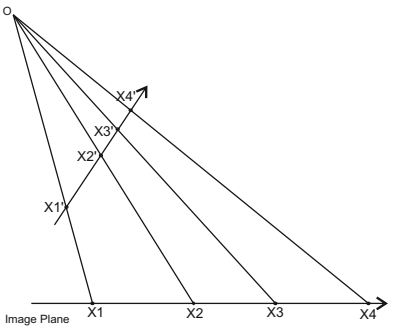

Figure 5: Cross Ratio

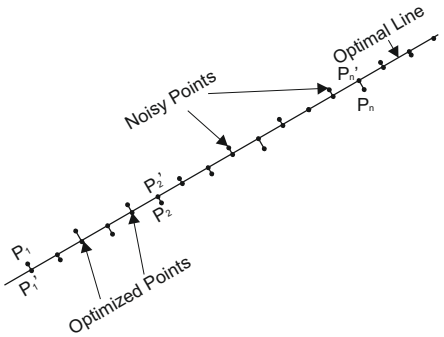

Figure 6: Noisy image points and its projection on best fit line

In homogenous coordinates, (wu, wv, w) and $(u, v, 1)$ represent the same point. The factor w is merely a scaling quantity. The parameter α is defined as the ratio of scaling factors for two different points. It thereby provides wisdom about perspectivity. α can be evaluated from Eq. 10.

$$\alpha = \frac{w_2}{w_1} = \frac{(u_3 - u_1)(\lambda - 1)}{(u_3 - u_2)\lambda} \tag{12}$$

The vanishing point is the image of a point lying at infinity on line L1. This point (let us say is D) in the world frame will have a distance ratio, $(\lambda' = \frac{d_{A,D}}{d_{A,B}})$ of ∞. To get the VP, we substitute this value of λ' in Eq. 11. Through algebraic manipulation, we get,

$$(VP_x, VP_y) = \left(\frac{\alpha u_2 - u_1}{\alpha - 1}, \frac{\alpha v_2 - v_1}{\alpha - 1} \right) \tag{13}$$

An interesting phenomenon can be observed if we consider an image with zero perspective. For such an image α is equal to 1 since the scaling factors will be same for both the points. VP for such an image will be located at ∞ (Eq. 13). This is to be expected, since VPs arise only due to perspective in the image.

3.2 Proof by Invariance

Property: The cross ratio (χ) is invariant under projective transformation. χ is expressed as,

$$\chi = \frac{d_{13} d_{24}}{d_{12} d_{34}} \tag{14}$$

where $d_{i,j}$ represents distance between points i, j as shown in Fig. 5.

In our current formulation let us assume that a fourth point V is lying on line L1 at an infinite distance along with A, B and C. V' projected on the image plane from V, will represent the vanishing point. Using Eq. 5, cross ratio in world frame is given by,

$$\chi = \lim_{V \to \infty} \frac{d_{AB} d_{CV}}{d_{AC} d_{BV}} = \frac{1}{\lambda} \lim_{V \to \infty} (1 + \frac{d_{CB}}{d_{BV}}) = \frac{1}{\lambda} \tag{15}$$

In the image frame, let the coordinates of V' be given by Eq. 13, We write the cross ratio as,

$$\chi' = \frac{\sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2}}{\sqrt{(u_1 - u_3)^2 + (v_1 - v_3)^2}} \times \frac{\sqrt{(\frac{u_3 - u_3\alpha - u_1 + u_2\alpha}{1-\alpha})^2 + (\frac{v_3 - v_3\alpha - v_1 + v_2\alpha}{1-\alpha})^2}}{\sqrt{(\frac{u_2 - u_1}{1-\alpha})^2 + (\frac{v_2 - v_1}{1-\alpha})^2}} \tag{16}$$

Substituting the value of α from Eq. 12 and simplifying algebraically, we get $\chi = \chi'$. This shows that the cross ratio of four points is invariant under projection and our VP estimates are accurate.

3.3 Tackling Noise

In the presence of noise, the performance of image processing techniques may get degraded. If the location of those three collinear points is not known precisely, errors will creep in to the VP estimates. To reduce this sensitivity we incorporate a least squares based optimization method. The idea behind this method is to draw a least square fit line from the selected three points to find the direction. Then orthogonally project these points on this line. These new points are used in place of earlier noisy data see Fig. 6.

4 RESULTS

Experiments were performed to validate the 3-Point (3-PVP) method. Robustness of the method and its computational efficiency are investigated through simulations. All simulations are performed in MATLAB® environment. Simulations were performed on a PC with i5 processor (3.2 GHz, 64 bit) and 4 Gb RAM.

[4] and [3] focus on vp estimation in urban/man-made environment where determining distance ratio will be difficult and will have to be separately estimated. Hence, we compare our results with LSVP method described in Section 2.

 ISBN 978-80-86943-75-6

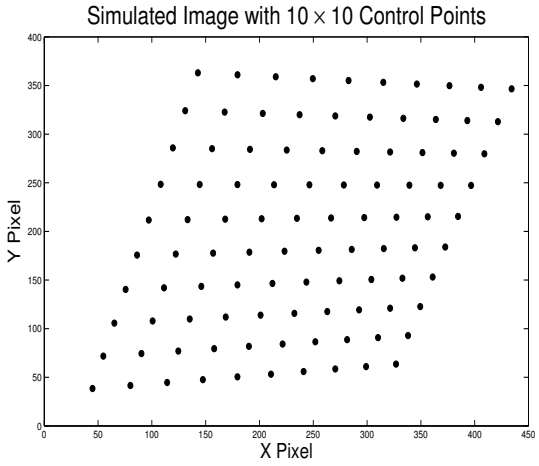

Figure 7: Simulated Pattern of 10×10 control points

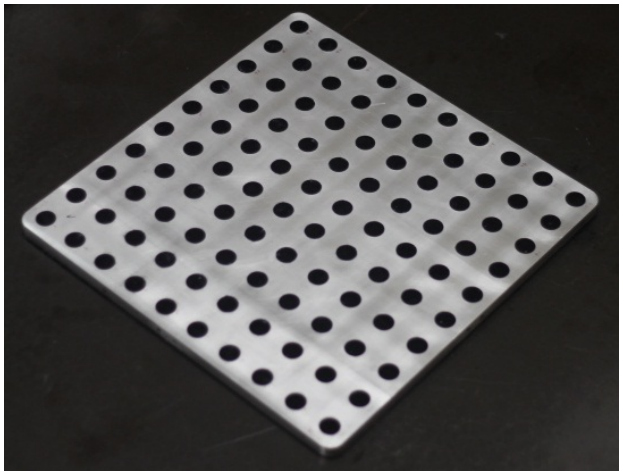

Figure 8: Metal fixture with 10×10 control points

4.1 Simulation

We here simulate perspectivity transform and generate synthetic images. A target pattern with 10×10 evenly distributed points is simulated as shown in Fig. 7. Two sets of parallel lines can be drawn in X and Y direction. This pattern is projected on the synthetic image-plane using a simulated camera. Properties are tabulated in Table 1.

Absolute ground truth can be found out for simulated images and hence it can be a great tool to validate estimation method. Since, we are simulating perspective, the camera matrix $\mathbb{P}$ is known to us and homogenous coordinates (of infinity point along the line) are known to be $[1,0,0,0]$.

Synthetic images provide us with an unique opportunity to add Gaussian noise and analyse robustness. Gaussian noise $\mathcal{N}(0,0.1px)$ is added to each projected point on the pattern. This new noisy image is given as input to the VP estimation algorithm. Two vanishing points are estimated in each image, represented by VP1 and VP2. We perform 100 Monte-Carlo runs for both methods.

Accuracy and Robustness

We compare our results with least square approach. 3-PVP provides accurate VP estimates. This is seen from the fact that 3-PVP estimates are closer to the ground truth. The mean error and standard deviation of error are also lower for 3-PVP as compared to LSVP. The mean errors are negligibly small as compared to VP estimate for both methods. Standard deviation of error is approximately one third the value of LSVP. Results are tabulated in Table 4.

Noise with std of 0.1 px was used to study robustness. 3-PVP method is shown to be robust to image noise. Euclidean norm of error is much higher for LSVP as compared to 3-PVP. Error norm for LSVP is approximately three times that of 3-PVP. Error values are tabulated in Table 3

Parameter	Value
Focal Length	50 mm
Principal Point	(360,240) px
Skew Factor	0
Scale Factor	1
Orientation Vector	$[0^0\ 40^0\ 30^0]$
Translation Vector	$[800,-1200,300]$ mm
Image Resolution	720×480

Table 1: Simulated camera parameters

Fig. 9 and Fig. 10 shows the ground truth location of VP and the spread of estimated VPs using both methods. The VPs estimated by LSVP have more deviation from the true value. Fig. 11 represents the error in x and y direction along with error norm for the Monte-Caro runs. 3-PVP method shows lesser error as compared to LSVP.

Computational Cost

Monte-Carlo runs also can indicate the computational cost of the algorithm. The time taken by both methods are tabulated in Table 2 for 100, 1000 and 10000 runs. We can observe that the speed of 3-PVP is approximately ten times faster. LSVP method involves firstly forming least square lines and secondly finding their intersection. 3PVP on the contrary employs an algebraic relation and is hence fast. These simulations validate our method's accuracy, robustness and speed.

Number of Runs	3-PVP (s)	LSVP (s)
100	0.230184	1.27458
1000	1.5083	13.09765
10000	14.9408	132.65978

Table 2: Simulation Time

4.2 Experimental Results

The target used for validating the 3-P VP method is shown in Fig. 8. The coordinates of the center of circles are known with high degree of accuracy. These

 ISBN 978-80-86943-75-6

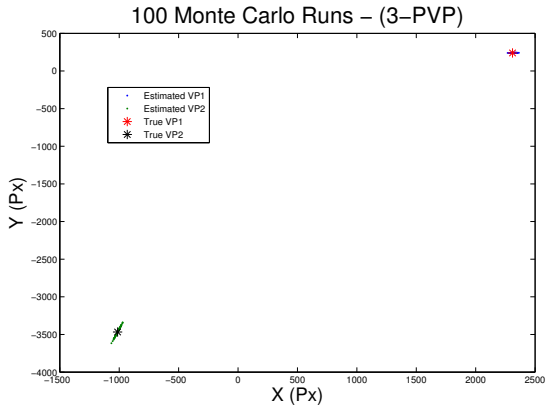

Figure 9: VP estimate spread for 3-PVP

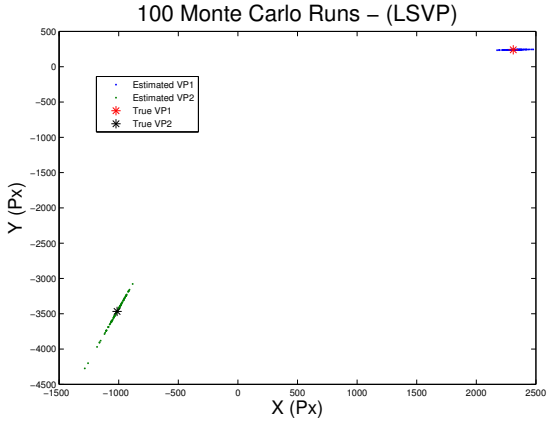

Figure 10: VP estimate spread for LSVP

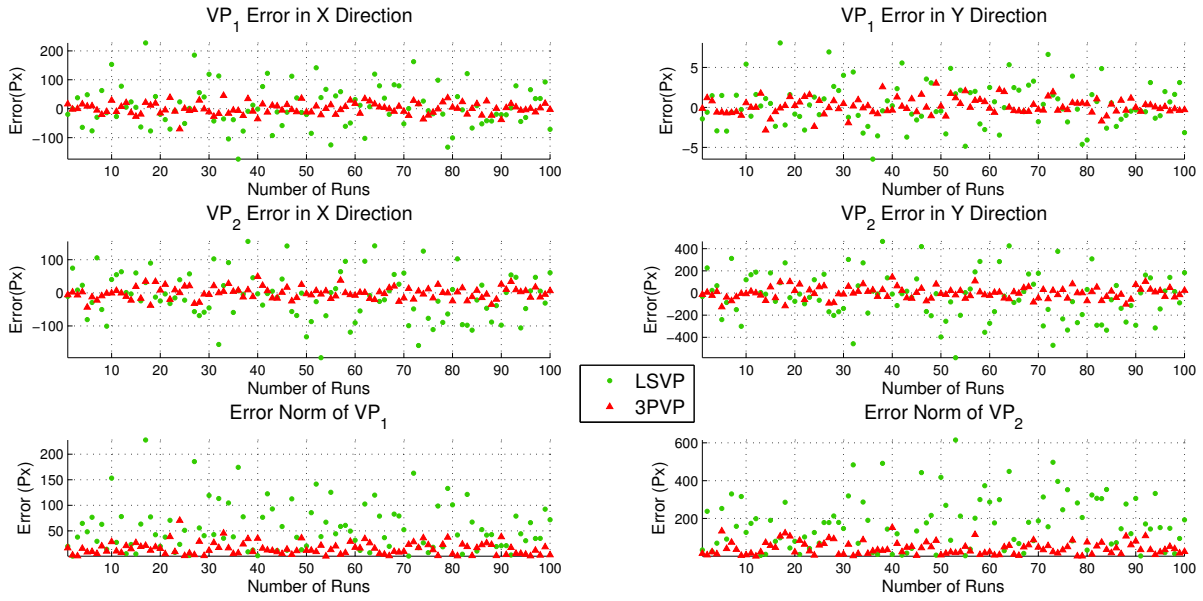

Figure 11: Error in VP Estimates (RMS, X & Y Direction) for 3-PVP v/s LSVP

points are planer in nature and form two sets of parallel lines. The image is captured using a Cannon EOS 1100 D camera with fixed effective focal length of 50 mm. Median filter has been used to remove noise from the image. A well-focused image of the target is processed in MATLAB to obtain centroids of the circular control points. Two vanishing points are obtained from each image. The vanishing points obtained by the 3-p strategy are compared with results from LSVP method.

3-PVP and LSVP are used on three images and their VPs are estimated. The results show that both the methods work effectively with the current image. The distance between results from both methods is shown to be of the order of 10^{-08} or lower. This also shows that in the absense of noise both methods will converge to the same estimate. VP estimates for those three images are tabulated in Table 5.

The advantages of our method are,

- *Tackling of radial distortion*: Our algorithm gives us the freedom to select the three points, which can be selected such that they lie in the middle of the image. Radial distortion effects are negligible near the center.
- *Handling defocused images*: Partial defocusing in images can lead to large erros in feature extraction. We can select required three points in such a way that you avoid defocused parts of the image.
- *Independent of Parallel Lines*: Errors also creep in when the given set of lines is not perfectly parallel. We do not need parallel lines and hence are not prone to errors.
- *Fast and Robust*

 ISBN 978-80-86943-75-6

VP	True VP (px)	3-PVP Error Norm		LSVP Error Norm	
		Mean (px)	STD (px)	Mean (px)	STD (px)
VP_1	(2310.1, 240)	15.59	11.81	50.36	38.09
VP_2	(-1013.1, -3467.3)	52.37	37.75	166.2	146.8

Table 3: Comparison of mean and std of error norm in VP estimation using 3-PVP and LSVP

VP	True VP (px)	3-PVP Error		LSVP Error	
		Mean (px)	STD (px)	Mean (px)	STD (px)
$VP_{1,x}$	2310.1	15.5410	11.8248	50.3179	38.073
$VP_{1,y}$	240	0.6822	0.5821	1.9273	1.4071
$VP_{2,x}$	-1013.1	16.8240	12.1201	52.6534	46.5807
$VP_{2,y}$	-3467.3	49.5547	35.8124	157.5836	139.2167

Table 4: Comparison of 3-PVP and LSVP estimates and error analysis

Image	VP	3-PVP (px)	LSVP (px)	$\| \varepsilon \|$(distance)
Image 1	VP_1	(279.45, -1179.07)	(279.45, -1179.07)	6 E -11
	VP_2	(-36027.08, 4453.91)	(-36027.08, 4453.91)	1 E -08
Image 2	VP_1	(2601.13, -1539.28)	(2601.13, -1539.28)	2 E -10
	VP_2	(-1778.97, -859.70)	(-1778.97, -859.70)	1 E -10
Image 3	VP_1	(1301.67, -1121.90)	(1301.67, -1121.90)	1 E -10
	VP_2	(-4198.81, -644.39)	(-4198.81, -644.39)	1 E -10

Table 5: Vanishing Points of real images using 3-PVP and LSVP

5 CONCLUSION

The 3-PVP method is based on single point perspectivity. Three collinear points and their distance ratio in the world frame characterize perspectivity in the direction of that line. It provides us with a non-iterative, closed-form solution. It is proved to be accurate and robust. VP estimation was performed on simulated images with a gaussian noise $\mathcal{N}(0, 0.1)$. It provides VPs with approximately one third the error norm and a smaller standard deviation as compared to LSVP. 3-PVP method is shown to be computationally cheap. Experimental results show that in the absence of noise, 3-PVP method and LSVP converge to the same value (accurate estimation) albeit with much lesser computational time. It has promising future in applications which require high accuracy VP estimation.

ACKNOWLEDGMENT

We gratefully acknowledge the help and guidance provided by Mr. R S Chandrasekhar, RCI.

6 REFERENCES

[1] Hartley, Richard and Zisserman, Andrew: MultipleView Geometry in computer vision, Cambridge UniversityPress (2000)

[2] Forsyth, D.,Ponce A., Computer Vision: A Modern Approach, 2nd Edition: Prentice Hall, ch 16.1, pp. 437-439 (2011)

[3] Tardif, J.P., Non-iterative Approach for Fast and Accurate Vanishing Point Detection, 12th IEEE International Conference on Computer Vision, Kyoto, Japan, pp. 1250-1257 (2009)

[4] Barinova, O. et. al., Geometric Image Parsing in Man Made Environments. 11th European Conference on Computer Vision, pp. 57-70 (2010)

[5] Denis, P., Elder, J. H., Estrada, F. J.: Efficient Edge-based Methods for Estimating Manhattan Frames in Urban Imagery. ECCV 2008, Part II, LNCS 5303, pp. 197-210 (2008)

[6] Tsai, R.Y.: A versatile camera calibration technique forhigh accuracy 3D machine vision metrology using offthe-shelf TV cameras and lenses. IEEE J. Robotics Automat.,Vol. RA-3, No. 4, pp. 323-344(1987)

[7] Grammatikopoulos, L., Karras, G., Petsa, E., Camera calibration combining images with two vanishing points. Int. Arch. of Photogrammetry, Remote Sensing and Spatial Information Sciences, 35 (Part 5), pp. 99-104 (2004)

[8] Li, B. et. al.: Simultaneous vanishing point detectionand camera calibration from single images. Proceedingsof the 6th international conference on Advancesin visual computing, pp. 151-160, Volume Part II (2010)

[9] Caprile, B., Torre, V., Using vanishing points for camera calibration. Int. Journal of Computer Vision, 4(2), pp. 127-140 (1990)

 ISBN 978-80-86943-75-6

[10] Lee, D. H., Jang K. H., and Jung, S. K.: Intrinsic Camera Calibration Based On Radical Center Estimation. The 2004 International Conference on Imaging Science, Systems, and Technology, USA, pp. 7-13, (2004)

[11] He. B.W., Li Y.F., Camera calibration from vanishing points in a vision system, Optics and Laser Technology, Volume 40, pp. 555-561(2008)

[12] Orghidan, R. et al.: Camera calibration using two or three vanishingpoints, Proceedings of the Federated Conference onComputer Science and Information Systems, pp. 123-130 (2012)

[13] Fong, C.K.: 3D object reconstruction from single distortedline drawing image using vanishing points. Proceedings of ISPACS 2005 pp. 53-56 (2005)

[14] Parodi, P. and Piccioli, G.: 3D shape reconstructionby using vanishing points. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18,no. 2, pp. 211-217 (1996)

[15] Schuster, R., Ansari, N. and Bani-Hashemi,A.: Steering a Robot with Vanishing Points. IEEE-Transactions on Robotics and Automation,Vol. 9, NO. 4, pp. 491-498 (1993)

[16] Wang, Y.: An efficient algorithm for UAV indoorpose estimation using vanishing geometry. 12th IAPRConference on Machine Vision Applications, Japan, pp. 361-364 (2011)

 ISBN 978-80-86943-75-6