

 Open access • Proceedings Article • DOI:10.5244/C.7.53

An Optimised Vanishing Point Detector — [Source link](#)

Phil Palmer, A. Tai

Published on: 01 Jan 1993 - British Machine Vision Conference

Topics: Vanishing point, Hough transform, Line segment, Line (geometry) and Kernel (image processing)

Related papers:

- [New automatic vanishing point detection algorithm](#)
- [Road Detection Using a New Method for Vanishing Point Detection Based on Hough Transform](#)
- [A New Vanishing Point Detection Algorithm Based on Hough Transform](#)
- [A New Vanishing Point Detection Algorithm for Orthogonal Directions](#)
- [Determining vanishing points using Hough transform](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/an-optimised-vanishing-point-detector-2wf6mxwn9j>

An Optimised Vanishing Point Detector

P. Palmer & A. Tai

Dept. of Electronic and Electrical Engineering,
University of Surrey,
Guildford, Surrey GU2 5XH, United Kingdom.

Abstract

In this paper we use line segments from a Hough transform algorithm to locate vanishing points in an image. The line parameters have already been determined to high accuracy, and the purpose of this paper is to present a scheme for locating the vanishing points from the line intersections which takes full advantage of this accuracy. We present a natural generalisation of the usual accumulator method which incorporates statistical hypothesis testing to account for the effects of noise and errors in line segment parameters. Using this smooth voting kernel in the accumulation process, we have developed an optimisation scheme as a post-process to remove sampling errors in the vanishing point accumulator. We demonstrate the improvement in the results using synthetic imagery for which ground truth is known. We then demonstrate the algorithm on two images of outdoor scenes. The first is a road scene for which we determine vanishing points for a building in the street, and the second is an infra-red image of a runway as seen from an approaching aircraft.

1 Introduction

Many natural scenes possess sets of lines which are parallel in the 3D world. When these lines are projected under perspective projection, the lines project onto the image plane as lines which have a common intersection point - the vanishing point. If, therefore, we can locate vanishing points in an image, and the lines which intersect at it, then we have obtained some 3D information about this set of lines which will prove useful when interpreting the world scene from the 2D image [3]. Sets of lines in the 3D world which are coplanar project onto lines whose vanishing points lie along a straight line in the image plane - the vanishing line. Determining two or more vanishing points which lie on a vanishing line gives a very strong cue about the true geometric relationships between objects in the image scene. Aerial images are particularly susceptible to this approach as all objects in the image can be assumed to lie on the ground plane.

To locate vanishing points in an image Barnard proposed the projection of image lines onto the Gaussian sphere [1] [5] [11]. A line segment on the image plane and the centre of the camera lens define a plane in 3D on which the 3D line must also lie. The Gaussian sphere is centred on the camera lens and has unit radius. The plane containing the 3D line and its image cuts this sphere in a great circle. If the image contains a number of parallel lines in the 3D world, then the great circles generated from their projections on the image plane will coincide at a point on the Gaussian sphere, from which the location of the vanishing point can be determined.

The surface of the Gaussian sphere can be mapped by two angles, and we can divide this two angle parameter space (the Hough plane) into equally spaced

bins and accumulate votes over these bins. This accumulation can be performed by allowing each line segment to vote for all bins around the great circle of the sphere, or by considering line pairs in the original image and determining which bin on the Gaussian sphere this location maps to. Which of these two methods is better to use depends upon how many accumulator bins and how many line segments are involved.

The problem with this accumulation method is that the location of the vanishing points is poorly determined due to the sampling errors of the two angles on the Gaussian sphere. A crude way of ensuring better accuracy is to partition the Hough plane into finer bins. The improvement in accuracy obtained, however, is paid for by an increase in memory requirement and computational load. Another alternative is to compute the vectors pointing towards the intersection of line segments in the image plane using a series of cross-product operations [5]. The actual values of the two angles are then maintained for comparison using an arc distance as a metric. Although this allows one to locate vanishing points to a higher accuracy, the computation of the vectors pointing at the intersection points has transformed the $O(N)$ problem to an $O(N^2)$ one.

Another approach is to use a pyramidal approach to the Hough accumulation as used in the Fast Hough Transform algorithm [4]. A patch of the Gaussian sphere is recursively subdivided from a coarse to fine resolution [11]. Detailed experimental studies, however, suggest that to ensure the detection of all level features in a hierarchical approach may render the technique computationally inferior to a standard Hough implementation.

All these methods rely on some form of accumulation of line pair junctions using a parameterisation of the Gaussian sphere. There are several shortcomings associated with such an approach. The most significant problem is that no consideration is given to the accuracy with which the detected line parameters have been determined. Positional and orientational errors cause incorrect intersection points to be formed which reduce the strength of the 'true' peaks and give rise to spurious vanishing points. Brillault-O'Mahony [2] has recently designed an isotropic accumulator space where the probability of erroneous vanishing point detection is uniformly distributed throughout all the bins.

In this paper we propose an adaptation of the accumulation method which takes proper account of the effects of noise and errors in line segment parameters. This is achieved by adjusting the way votes are cast during the accumulation. We further introduce a post-processing optimisation method that overcomes the shortcomings described above without incurring high costs due to high sampling frequency of the parameter space. In the next section we describe the method and give estimates of the errors in vanishing point location due to under-sampling. In the third section we show results of our optimised vanishing point detector and show that we can overcome these sampling inaccuracies at cheap computational cost. We demonstrate the accuracy achieved using synthetic imagery and apply the method to two outdoor scenes. Finally in section 4 we discuss the method in the general context of vanishing point detection.

2 Optimised Vanishing Point Algorithm

Our vanishing point algorithm uses as input a set of line segments provided by a Hough transform algorithm. The Hough transform was chosen as it is a very robust method for detecting straight line segments, and a fast algorithm has been developed [6]. The input to this routine comes from an edge detector using subpixel

accuracy [12] and an optimising filter [9]. In this way we minimise the errors associated with the edge detection phase of the processing. The errors associated with the estimated line parameters from the Hough algorithm are dominated by the sampling frequency in the Hough parameter space. As a result, algorithms have been developed to remove this uncertainty in the line parameters. One of these methods uses an optimisation scheme [7] and another uses focus of attention inside the Hough algorithm to improve the accuracy on identified groups of lines in the image [8].

In this paper we describe an algorithm for vanishing point detection which preserves the accuracy maintained at the lower levels of processing. This algorithm employs an accumulator over the two angles defined by:

$$\alpha = \tan^{-1} \left(\frac{y_p}{x_p} \right) \quad (1)$$

and

$$\beta = \tan^{-1} \left(\frac{r_0}{r_p} \right) \quad (2)$$

where the vanishing point is located at (x_p, y_p) and r_p is the radius at which this vanishing point is located in the image plane, taking the origin at the point where the optical axis cuts the image plane. The quantity r_0 is a parameter which is chosen arbitrarily. For vanishing points which lie near the focal axis of the camera, the angle α is very uncertain and the method fails. We therefore use the radius r_0 to exclude any candidate vanishing points which lie within this circle. This is achieved by only accumulating over angles β in the range $-\pi/4$ to $\pi/4$ ($r_p \geq r_0$). The range of the angle α is $-\pi$ to π .

An accumulation is performed over this parameter space using all possible pairs of lines from the Hough algorithm. The computational cost is reduced by eliminating pairs of lines which intersect near the centre of the image and lines which are closely parallel. The accumulator is then passed through a routine which suppresses all the non-maximal peaks and then is thresholded. The uncertainty in the two angles α and β and hence in the location of the vanishing point is dominated by the sampling frequency of the parameter space. In our algorithm we use 150 bins in each of the two parameters. This corresponds to an uncertainty of 0.04 radians in α and 0.01 radians in β . From (1) and (2) we find the location of the vanishing point to be:

$$x_p = r_0 \cot \beta \cos \alpha \quad \text{and} \quad y_p = r_0 \cot \beta \sin \alpha \quad (3)$$

The uncertainty in the angles produces an uncertainty in the location of the vanishing point of Δx_p in x_p and Δy_p in y_p , where:

$$\Delta x_p = y_p \Delta \alpha + x_p \left(\frac{r_0^2 + r_p^2}{r_0 r_p} \right) \Delta \beta \quad (4)$$

and

$$\Delta y_p = x_p \Delta \alpha + y_p \left(\frac{r_0^2 + r_p^2}{r_0 r_p} \right) \Delta \beta \quad (5)$$

It is clear from these expressions that the errors in the vanishing points are small if x_p and y_p are comparable and r_p is comparable to r_0 . The problem is that we

cannot fine tune r_0 without knowing roughly how far out the true vanishing point is beforehand since we discard all candidate vanishing points within radius r_0 . As a result the second term in equations (4) and (5) are amplified by roughly a factor of r_p/r_0 . Suppose the vanishing point is located at approximately $x_p \sim y_p \sim 100$ pixels and $r_p = 5r_0$, then the errors in the location of the vanishing point, just due to sampling errors of the parameter space, would be approximately 5%. If, however, the vanishing point were at $x_p \sim y_p \sim 1000$ and r_0 unchanged, then the error would be 27 %. Hence the sensitivity of the test depends strongly on the choice of r_0 .

Since we do not know *a priori* how to choose r_0 , we have adopted a scheme which allows us to remove the sampling error after accumulation has taken place. In this way we can reduce the uncertainties $\Delta\alpha$ and $\Delta\beta$ in the above formulae to their smallest possible values associated with the accumulated errors from the edge detector and Hough line finders, which have already been optimised. For each pair of lines found in the image, we determine an intersection point and find the corresponding values of α and β . Instead of incrementing the accumulator bin in which this point resides in the usual way, we allow for the uncertainties in the line parameter estimates and spread the vote of the line pair over a number of accumulator bins, dependent upon how long the line segments are, and how far from the ends of these segments the point of intersection was found to be. The vote that the line pair contributes to the accumulator was calculated from a smooth voting kernel which peaks at unity when the intersection point coincides exactly with the centre of the accumulator bin, and falls smoothly to zero. To determine the shape of this voting kernel we view the accumulation process in terms of hypothesis testing [10]. The values of the two angles at the centre of the bin is the hypothesis and the intersection points of line pairs provide the support for the different hypotheses being tested. The shape of the voting kernel is determined by the requirement that the hypothesis has a high probability of being accepted (above threshold) when it coincides with a true vanishing point in the image, and this probability must fall rapidly to zero as the hypothesis differs from the true vanishing point location. A detailed analysis of this is given by Palmer *et. al.* [6], with the result that the form of the voting kernel was chosen to be:

$$K(\delta\alpha, \delta\beta) = F\left(\frac{\delta\alpha}{K_\alpha}\right)F\left(\frac{\delta\beta}{K_\beta}\right) \quad (6)$$

where

$$F(x) = 1 - 2x^2 + x^4 \quad (7)$$

provided $|x| < 1$ or zero otherwise, and K_α and K_β are the predetermined widths over the parameter space that each line pair is allowed to vote. The values of $\delta\alpha$ and $\delta\beta$ are computed as the difference between the angles equivalent to the intersection point and the angles at the centre of the current bin in the accumulator.

The advantage of introducing this extended voting kernel in the vanishing point accumulator is that the final accumulator is a smooth function of position in the parameter space with localised peaks associated with the vanishing points in the image. At this stage we only have the values of the accumulator at the regular grid locations of parameter space. We now employ a hill climbing algorithm to search for these peaks in the accumulator function to sub-bin accuracy. By keeping track of which line pairs voted for each accumulator bin above threshold, the number of line pairs associated with any individual peak is small. The cost of re-evaluating

Truex	Truey	VPx	VPy	Optx	Opty
300.5	406.44	309.43	413.89	299.99	405.93
-43.5	133.42	-43.95	135.20	-44.00	132.84
-171.5	31.83	-174.21	36.76	-172.01	31.33

Table 1: Co-ordinates of the three vanishing points determined from ground truth. In the middle two columns are the results from the VP detector before optimisation. There is no noise so this error just represents sampling errors. The last two columns are the locations after optimisation.

the accumulator over these line pairs as we climb the hill is therefore not great and enables us to accurately locate the peak to within some small tolerance.

The hill climbing method we employed is based on making a surface fit to the accumulator function up to quadratic terms in each of the two parameters. By using the first and second derivatives of the kernel, which can easily be computed from (6) and (7), the coefficients of this fit can be determined, and then by differentiating the fit function, the location of the peak in the fitting surface found. We then move to this peak location and re-evaluate the accumulator and its derivatives until the difference between our current location in parameter space, and the estimated location of the local peak coincide within some tolerance.

3 Experimental Results

In order to test our algorithm for vanishing point detection we created an artificial image containing three groups of seven lines. Each of these lines corresponded to a set of 3D parallel lines which would be coplanar in the 3D world. The orientation of this plane is known, and the locations of the three vanishing points on the image plane are also known. In this way we could test the effects of sampling errors and the accuracy of our optimisation scheme. The locations of the three vanishing points are given in table 1, along with the ground truth values. Also presented are the values obtained from the vanishing point detector before optimisation. We see that the results were significantly improved by the optimisation scheme. The amount of cpu time required to perform the optimisation on all these vanishing points was less than 0.1 secs on a Sparc2.

We show the locations of the three vanishing points found to show how close to colinear they are. We determined from these three points the equation of this vanishing line by a least squares fit and hence were able to estimate the orientation of the 3D plane on which these lines would have sat. The orientation of this plane was determined in terms of the three components of the unit vector orthogonal to the plane. From the least squares fit line we obtained (0.5002, 0.6303, 0.5937) for this vector. Ground truth for this orientation was (0.5, 0.63, 0.594). Using the non-optimised values we obtain for the plane orientation (0.4972, 0.6273, 0.5993).

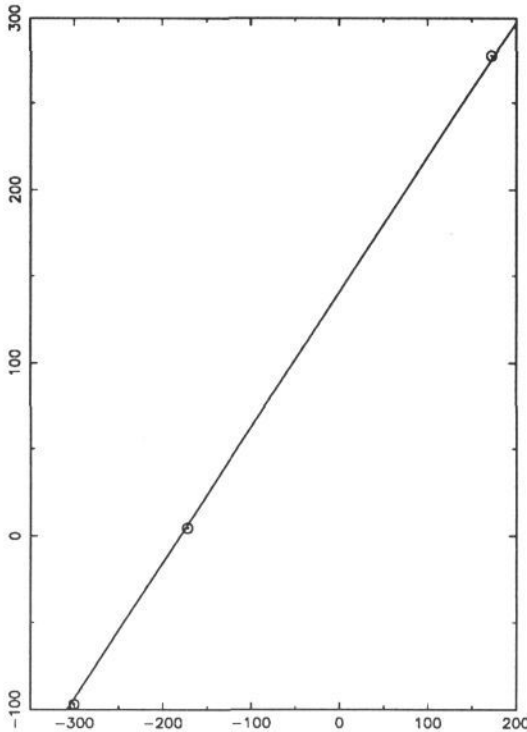


Fig 1: Least Square fit to 3 vanishing points giving the vanishing line.

We also ran the image through the edge detector and line finder to see how large the errors were due to these two levels of processing. The values for the locations of the three vanishing points are given in table 2. For comparison the ground truth has been duplicated in the first two columns and the locations before and after optimisation in the next four columns. The first vanishing point has been located very accurately and the largest error in the co-ordinates of the second point is roughly 6.4%. The last point, however, has a larger error of 20% in the y co-ordinate. This large error was traced to the edge detection phase of the processing. We tested that the optimisation procedure was correct by making an exhaustive search and reproduced the same values as in table 2. Apart from this exceptional error, the errors in the other co-ordinates are roughly 6%. This table demonstrates that the effects of optimisation have significantly improved the location estimates of the three vanishing points.

Having demonstrated the improvement obtained on a synthetic image for which ground truth was known, we next applied the algorithm to a real outdoor scene shown in figure 2. Using lines obtained for the balconies on the front of the building in the centre of the image, we were able to determine a vanishing point at (286.3, 105.2). Using the optimisation scheme the location of this vanishing point had moved to (286.1, 108.0). We see that in this particular case the x location of the vanishing point coincided very closely with one of the bins although the y co-ordinate shifted by 3%. Averaging over all the vanishing points showed that

Truex	Truey	VPx	VPy	Optx	Opty
300.5	406.44	297.15	394.5	299.8	403.1
-43.5	133.42	-43.95	135.21	-46.3	133.92
-171.5	31.83	-191.77	16.65	-183.2	25.3

Table 2: Co-ordinates of the three vanishing points determined from the Hough line segments. These results include the effects of noise. In the middle two columns are the results from the VP detector before optimisation and the last two columns are the locations after optimisation.

the optimisation scheme has a 0.1 sec cpu overhead per peak in the accumulator.

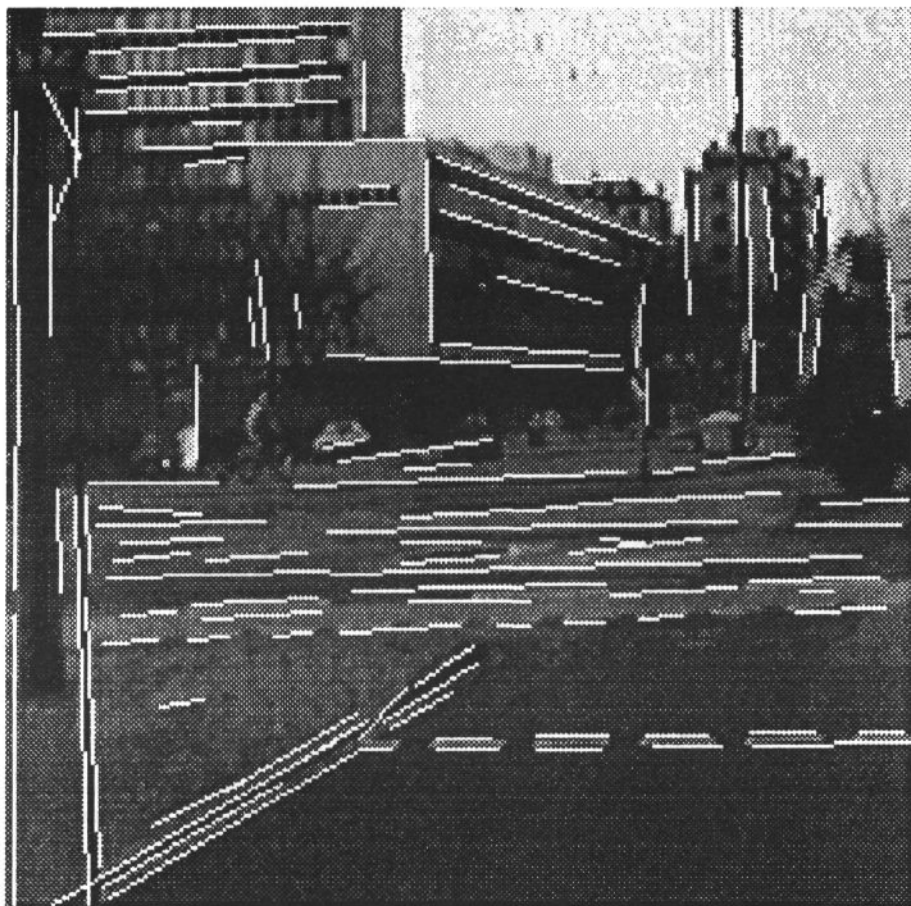


Fig 2: Outdoor road scene with line segments used for vanishing point detection.

As a final demonstration of the effects of optimisation on the location of the vanishing point, we applied the scheme to the image in figure 3. This shows an infra-red image of a runway taken from an approaching aircraft. There is an important vanishing point obtained from the sides of the runway, which was found

to be at (141.2,2.4) before optimisation and (139.5,2.6) afterwards.



Fig 3: Runway seen from an approaching aircraft. Vanishing point is determined from the sides of the runway.

4 Discussion

We have presented a modified form of the standard accumulation technique for locating vanishing points in an image using pre-detected straight-line segments. The method we have developed allows for uncertainties in the line parameters used and is based on a rigorous hypothesis testing analysis. With this method we have been able to produce a post-processing optimisation level within the detector to remove the errors in vanishing point location due to undersampling the two dimensional parameter space. This optimisation scheme is fast and efficient and allows us to reduce the number of bins in the accumulator array and so speed up the accumulation phase of the detection process. We experimented with reducing the number of accumulator bins and found we could still reproduce the exact locations of the vanishing points by reducing the number of bins to 100 in each parameter. This increased the speed of the algorithm by more than a factor of

two. The adaptation of the standard vanishing point detection process is easy to encode.

The resulting algorithm can locate vanishing points accurately and efficiently. The main problem remains in determining which line segments in the image should be associated together with a common vanishing point. If the optimisation scheme uses a set of lines of which some are not 3D parallel, then the improvement in accuracy will not, in general, determine the correct location for the vanishing point but a compromise between the different line intersections. To overcome this difficulty requires some higher level knowledge of the image, as any group of lines which are 3D parallel will tend to produce peaks in the accumulator with other spurious line segments in the image. The hypothesis testing voting kernel we have employed in our algorithm is designed to reduce this problem, but clearly this is now the main source of error.

An alternative approach to vanishing point detection has been proposed by Tai *et. al.* [13]. This method has two main advantages - firstly it works when the image contains few 3D parallel lines. In such an image the accumulator method would produce very small peaks which would be difficult to detect on thresholding. The second main advantage is that it provides a direct measure of the confidence in the detected vanishing points as it takes full account of the uncertainties in the line parameters associated with it. The hypothesis testing accumulation we have described in this paper is a natural way to extend the accumulation method to take account of line segment errors, and since the final vote of each line pair can vary smoothly between zero and one, the height of the accumulator gives a measure of the support for the hypothesised vanishing point. This method therefore is a natural extension of the above method to images where enough parallel lines exist for the accumulation method to be adequate.

References

- [1] Barnard, S.T., "Methods for Interpreting Perspective Images", *Image Understanding Workshop*, Paolo Alto, California, 1982, 193-203.
- [2] Brillault-O'Mahony, B., "A Probabilistic Approach to 3D Interpretation of Monocular Images", Ph.D. Thesis, City University, London, 1992.
- [3] Haralick, R.M., "Using Perspective Transformation in Scene Analysis", *Computer Vision, Graphics & Image Processing*, **13**, 1980, 191-221.
- [4] Li, H., Lavin, M.A. & LeMaster, R.J., "Fast Hough Transform: A Hierarchical Approach", *Computer Vision, Graphics & Image Processing*, **36**, 1986, 139-161.
- [5] Magee, M.J. & Aggarwal, J.K., "Determining Vanishing Points from Perspective Images", *Computer Vision, Graphics & Image Processing*, **26**, 1984, 256-267.
- [6] Palmer P.L., Petrou M. & Kittler J., "A Hough Transform Algorithm with a 2D Hypothesis Testing Kernel", *CVGIP: Image Understanding*, accepted for publication.
- [7] Palmer, P.L., Kittler, J. & Petrou, M., "An Optimising Line Finder Using a Hough Transform Algorithm", *submitted to IEEE Pattern Analysis & Machine Intelligence journal*.
- [8] Palmer, P.L., Kittler, J. & Petrou, M., "Using Focus of Attention with the Hough Transform for Accurate Line Parameter Estimation", *submitted to Pattern Recognition journal*.
- [9] Petrou M. & Kittler J., "Optimal Edge Detectors for Ramp Edges", *IEEE Trans. in Pattern Analysis & Machine Intelligence*, **PAMI-13**, 1990, 483-491.
- [10] Princen J., Illingworth, J. & Kittler, J., "Hypothesis Testing: A Framework for Analysing and Optimising Hough Transform Performance", *to appear IEEE Trans. in Pattern Analysis & Machine Intelligence*, 1993.
- [11] Quan, L. & Mohr, R., "Determining Perspective Structures using Hierarchical Hough Transform", *Pattern Recognition Letters*, **9**, 1989, 279-286.
- [12] Spacek L.A., "Edge Detection and Motion Detection", *Image and Vision Computing*, **4**, 1986, 43-56.
- [13] Tai A., Kittler J., Petrou M. & Windeatt T., "Vanishing Point Detection", *British Machine Vision Conference*, Ed. D. Hogg & R. Boyle, Springer-Verlag, 1992, pp.109-118.

Acknowledgements

The author's wish to acknowledge support by the Procurement Executive, Ministry of Defence for this work.