International Journal of Computer Applications (0975 – 8887)
Volume 65– No.10, March 2013

Vanishing Point Detection by Clustering on the Normalized Unit Sphere

Qiang He
Department of Mathematics, Computer and Information Sciences
Mississippi Valley State University
Itta Bena, MS 38941

Chee-Hung Henry Chu
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504-4330

ABSTRACT
In this paper, we give a new vanishing point detection algorithm, called the normalized unit sphere. By normalizing homogeneous coordinates in the original image space, we transform image points onto a normalized unit sphere. Further, we transform straight lines in image space into circles on normalized unit sphere. As a result, the vanishing point detection is implemented by searching the intersections of circles on the normalized unit sphere. This algorithm not only bounds the search space but treats the finite vanishing points and the vanishing points at infinity with the same way. The experimental results on synthetic and real data show good performance of this algorithm.

Keywords
Vanishing Point Detection, Normalized Unit Sphere, Canny Edge Detector, Least-Square Method, K-means Method

1. Introduction
As we know that the parallel lines with the same direction, in scene space, will intersect in one common point in image space through perspective transformation. This point is called vanishing point. The vanishing point is possible at infinity as the image projections of parallel lines in scene space are still parallel. Vanishing points provide much useful information for 3D structure of a scene [4, 6, 7, 9, 11]. The vanishing points have been successfully applied to compute the internal parameters of a camera [2, 10]. We can also make affine rectification using vanishing points [3]. Two vanishing points define a line, which is called vanishing line. The affine rectification is realized by projecting the finite vanishing line into infinity.

There are two typical works on vanishing point detection. The classical work on vanishing point detection is from Barnard [1]. He introduced a Gaussian sphere, which is a unit sphere centering at the optical center of the camera. A straight line in the image becomes a great circle on the Gaussian sphere. The intersections of those great circles correspond to the intersections of straight lines in the image space. The intersections with mostly hittings would be detected as vanishing points. The original vanishing points in the image space can be recovered by triangle similarity transformation. The Gaussian sphere transforms the unbounded search space into a limited unit sphere. Finite vanishing points and vanishing points at infinity lie on the Gaussian sphere without difference. Secondly, given the assumption of the zero-mean Gaussian measurement noises, Liebowitz [5] introduced the maximum likelihood estimate (MLE) on the vanishing point detection. The MLE of the vanishing point is to minimize the sum of squared orthogonal distances between the fitted lines and the measured lines' end points. The solution can be achieved using the Levenberg-Marquart numerical algorithm [8].

There are two important problems for vanishing point detection. One is how to limit search space since vanishing points may be at infinity. The other is how to treat the vanishing points at infinity. Here, we introduce a normalization process on homogeneous coordinates such that the search space is bounded and the original vanishing points are very easy to be recovered.

2. Method Description
Those intersection points, where many straight lines meet, will be regarded as potential vanishing points. In general, the vanishing point detection consists of two steps: accumulation step and search step. In our accumulation step, we transform the homogeneous coordinates from infinite original image space into a limited unit sphere, called the normalized unit sphere. During the search of vanishing points, the K-means method will be applied to cluster the points on the normalized unit sphere. The clustering property from the original space is empirically preserved on the unit sphere. In our research, we cluster data on the normalized unit sphere based on Euclidean distance measure.

2.1. Line representation and their intersections and grouping
The non-homogeneous coordinate for an image point is

$$\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Then the corresponding homogeneous coordinate for this point is

$$\bar{\mathbf{x}} = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

such that $u = x/w$ and $v = y/w$.

A line $\mathbf{l}$ in homogeneous coordinate is $\mathbf{l} = [a \quad b \quad c]^{\mathrm{T}}$ so that a point $\mathbf{x}$ lying on it only if its homogeneous coordinate $\bar{\mathbf{x}}$ satisfies

$$\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{l} = [x \quad y \quad w]\begin{bmatrix} a \\ b \\ c \end{bmatrix} = ax + by + cw = 0$$

If there are two lines $\mathbf{l}_1 = [a_1 \quad b_1 \quad c_1]^{\mathrm{T}}$ and $\mathbf{l}_2 = [a_2 \quad b_2 \quad c_2]^{\mathrm{T}}$, then their intersection point is $\mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$, where $\times$ is the cross product between two vectors. If $\mathbf{l}_1$ and $\mathbf{l}_2$ are parallel, then the third element of intersection point $\mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$ is zero. This means that the intersection point $\mathbf{p}$ between $\mathbf{l}_1$ and $\mathbf{l}_2$ is at infinity.

In our work, we apply the least squares method to fit straight lines to discrete edge points, and represent straight lines using homogeneous coordinates. The intersection points are then found by cross products of lines in homogeneous coordinates.

2.2. Unit normalization on homogeneous points

We normalize the homogeneous coordinates of points so that they become unit vectors in 3-space. A homogeneous point $\bar{\mathbf{x}} = [x \quad y \quad w]^{\mathrm{T}}$ is normalized to

$$\bar{\mathbf{x}}' = \begin{bmatrix} x/r \\ y/r \\ w/r \end{bmatrix}$$

where $r = \sqrt{x^2 + y^2 + w^2}$.

Through this transformation, the original image points will be transformed onto a unit sphere, which we call the normalized unit sphere. The desirable property of our unit sphere is that it preserves the transformation from the original homogeneous coordinates to non-homogeneous coordinates. After vanishing points are found on this unit sphere, they can be mapped to the image plane by dividing the third element into the first two, which is the usual transformation from homogeneous coordinates to non-homogeneous coordinates. If the third element is 0, then the vanishing point is at infinity and the first two elements give a unit direction vector for those parallel lines generating the vanishing point. In Gaussian sphere, in order to recover the original vanishing points, we need to know focal length and do triangle similarity transformation.

Through unit normalization, the intersection points of line pairs are on the unit sphere. Hence, we can limit the search space on the unit sphere similar to the Gaussian sphere. In addition, we treat the finite vanishing points and the vanishing points at infinity in the same manner.

2.3. Search method

We use the K-means method to search the vanishing point on the normalized unit sphere. The means of clusters are detected as vanishing points.

In general, we adapt the K-means clustering to our search along the following principles.

1. *The clusters should be compact within a small circle.* This helps to remove outliers and improve the accuracy of detection. To do this, we compute the mean update as

$$\mathbf{u}_{\mathrm{n}}^{\mathrm{k}} = \frac{1}{N_k} \sum_{\|\mathbf{x}_{\mathrm{i}}^{\mathbf{k}} - \mathbf{u}_{\mathrm{o}}^{\mathrm{k}}\| < \rho} \mathbf{x}_{\mathrm{i}}^{\mathrm{k}}$$

   where $\mathbf{u}_{\mathrm{n}}^{\mathrm{k}}$ is the update mean for cluster k. $\mathbf{u}_{\mathrm{o}}^{\mathrm{k}}$ is the current mean for cluster k. ρ is the limit of the search area. N_k is the number of points within the search area for cluster k. That is, we only use the nearby points of the current means to compute new means.

2. *The clusters should be dense.* We reject the sparse clusters, which are always false vanishing points. In order to check whether a cluster is dense, we simply define the cluster acceptance probability as $\pi^k = \frac{N_k}{N}$, where N is the total number of points. If this probability is small, we reject this cluster and restart another guess.

3. Properties of the Normalized Unit Sphere

Here, let us discuss two properties of the normalized unit sphere. The first property for our unit sphere is that it still preserves the transformation from the original homogeneous coordinates to the non-homogeneous coordinates. The second property is that there exists a one-to-one point mapping between the original image space and the normalized unit sphere.

Proposition 1. The transformation from the original homogeneous coordinates to non-homogeneous coordinates is preserved for the normalized unit sphere.

Proof. Given a non-homogeneous point $\mathbf{x} = [u \quad v]^{\mathrm{T}}$ in the original space and its corresponding homogeneous point is $\bar{\mathbf{x}} = [x \quad y \quad w]^{\mathrm{T}}$, where $u = x/w$ and $v = y/w$. Then, the corresponding point on the normalized sphere is $\bar{\mathbf{x}}' = [x' \quad y' \quad w']^{\mathrm{T}}$, where $x' = x/|\bar{\mathbf{x}}|$, $y' = y/|\bar{\mathbf{x}}|$, $w' = w/|\bar{\mathbf{x}}|$, and $|\bar{\mathbf{x}}|$ is the length of vector $\bar{\mathbf{x}}$. We have $\frac{x'}{w'} = \frac{x/|\bar{\mathbf{x}}|}{w/|\bar{\mathbf{x}}|} = \frac{x}{w} = u$ and $\frac{y'}{w'} = \frac{y/|\bar{\mathbf{x}}|}{w/|\bar{\mathbf{x}}|} = \frac{y}{w} = v$. □

That is, the recovery of non-homogeneous coordinates from the original homogeneous coordinates is preserved for the normalized unit sphere. If we want to recover the detected vanishing points from the normalized unit sphere to the original image space, we just do it such like the transformation from a homogeneous coordinate to its non-homogeneous coordinate.

Next, let us see what a line $\mathbf{l} = [a \quad b \quad c]^{\mathrm{T}}$ in the original image space is transformed into on the normalized unit sphere. We have

$$\begin{cases} \bar{\mathbf{x}}^{\mathrm{T}}\mathbf{l} = ax + by + cz = 0 \\ x^2 + y^2 + z^2 = 1 \end{cases}$$

where $\bar{\mathbf{x}} = [x \quad y \quad z]^{\mathrm{T}}$ is homogeneous coordinate. If we consider the equation $\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{l} = ax + by + cz = 0$ with regard to x, y, z as free parameters, then $\mathbf{l} = [a \quad b \quad c]^{\mathrm{T}}$ defines a plane on the normalized unit sphere. This plane passes through the origin of the normalized unit sphere coordinate system. The intersection between this plane and the unit sphere is a circle, which is the original line transformed on the normalized unit sphere. The potential vanishing points will be those intersection points of many circles. In Gaussian sphere, the original lines are transformed into great circles and the potential vanishing points will be those intersection points of many great circles. However, Gaussian sphere is centered at the camera center. In order to recover the original vanishing points in image space, we need to know focal length and do triangle similarity transformation. In the normalized unit sphere, our original vanishing points can be easily recovered by dividing the third element to get back to the original image space. In addition, if the vanishing point is at infinity, then the third element is 0. The first two elements stand for a unit direction vector for those parallel lines generating the vanishing points.

Using this unit normalization, we can limit the search space on the unit sphere like Gaussian sphere. In addition, there is no difference between finite vanishing points and infinite vanishing points on the normalized unit sphere. That is, the normalized unit sphere treats the finite vanishing points and the vanishing points at infinity with the same way.

In order to see that our normalized unit sphere works, we also need to show that the uniqueness of point transformation exists between the original image space and the normalized unit sphere. That is, two different points from non-homogeneous coordinates are still two different points after they are transformed onto the normalized unit sphere. Vice versa.

Proposition 2. The uniqueness of point transformation (one-to-one mapping) exists between the original image space and the normalized unit sphere.

Proof. Given two non-homogeneous points $\mathbf{x}_1 = [u_1 \quad v_1]^{\mathrm{T}}$ and $\mathbf{x}_2 = [u_2 \quad v_2]^{\mathrm{T}}$ in the original image space and their corresponding homogeneous points are $\bar{\mathbf{x}}_1 = [x_1 \quad y_1 \quad w_1]^{\mathrm{T}}$ and $\bar{\mathbf{x}}_2 = [x_2 \quad y_2 \quad w_2]^{\mathrm{T}}$, where $u_1 = x_1/w_1$, $v_1 = y_1/w_1$, $u_2 = x_2/w_2$, and $v_2 = y_2/w_2$. The corresponding points on the normalized unit sphere is $\bar{\mathbf{x}}'_1 = [x'_1 \quad y'_1 \quad w'_1]^{\mathrm{T}}$ and $\bar{\mathbf{x}}'_2 = [x'_2 \quad y'_2 \quad w'_2]^{\mathrm{T}}$, where $x'_1 = x_1/|\bar{\mathbf{x}}_1|$, $y'_1 = y_1/|\bar{\mathbf{x}}_1|$, $w'_1 = w_1/|\bar{\mathbf{x}}_1|$, $x'_2 = x_2/|\bar{\mathbf{x}}_2|$, $y'_2 = y_2/|\bar{\mathbf{x}}_2|$, $w'_2 = w_2/|\bar{\mathbf{x}}_2|$, and $|\bar{\mathbf{x}}_1|$ and $|\bar{\mathbf{x}}_2|$ are the lengths of vectors $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$, respectively.

Assume that $\mathbf{x}_1 \neq \mathbf{x}_2$, we want to show that $\bar{\mathbf{x}}'_1 \neq \bar{\mathbf{x}}'_2$.

By contradiction, assume that $\bar{\mathbf{x}}'_1 = \bar{\mathbf{x}}'_2$. Then from the above Proposition 1 that the transformation from the original homogeneous coordinates to non-homogeneous coordinates is preserved for the normalized unit sphere, we can obtain that $\mathbf{x}_1 = \mathbf{x}_2$. This is not true. So $\bar{\mathbf{x}}'_1 \neq \bar{\mathbf{x}}'_2$.

Assume that $\bar{\mathbf{x}}'_1 \neq \bar{\mathbf{x}}'_2$, we want to show that $\mathbf{x}_1 \neq \mathbf{x}_2$.

By contradiction, assume that $\mathbf{x}_1 = \mathbf{x}_2$. We keep the elements in extra dimension of homogeneous coordinates are always nonnegative. Then we set the corresponding homogeneous coordinates of $\mathbf{x}_1$ and $\mathbf{x}_2$ as $\bar{\mathbf{x}}_1 = [w_1\mathbf{x}_1 \quad w_1]^{\mathrm{T}}$ and $\bar{\mathbf{x}}_2 = [w_2\mathbf{x}_2 \quad w_2]^{\mathrm{T}}$ with $w_1 \geq 0$ and $w_2 \geq 0$. Obviously, $\bar{\mathbf{x}}'_1 = \bar{\mathbf{x}}'_2$. This is not true. So $\mathbf{x}_1 \neq \mathbf{x}_2$. □

Based on one-to-one mapping, every line intersection point in the original image space corresponds to one point on the normalized unit sphere. However, on the Gaussian sphere, every line intersection point in the original image space corresponds to two points, where these two points define a diameter. The number of clusters, on the Gaussian sphere, is doubled. Our method removes this redundancy.

4. Vanishing Point Detection Algorithm by the Normalized Unit Sphere

Our algorithm to detect vanishing points consists of the following steps:

(1). Compute the edge maps for given images using Canny edge detectors.

(2). Fit straight lines in the edge maps through least-square method and build a line set consisting of all lines $\mathbf{l}_1$, $\mathbf{l}_2$,…, $\mathbf{l}_M$, where M is the number of lines.

(3). Build a line intersection set from all cross products between two lines, that is, $\mathbf{l}_i \times \mathbf{l}_j$ for all $i \neq j$.

(4). Rectify all line intersections such that all intersections from the same vanishing point will converge into one cluster on the normalized unit sphere.

(5). Normalize all line intersections such that all points lie on the normalized unit sphere.

(6). Cluster points on the normalized unit sphere. The dense clusters will be detected as potential vanishing points.

(7). Compute the cluster means as vanishing points.

The intersection of two lines $\mathbf{l}_i$ and $\mathbf{l}_j$ for $i \neq j$, equals to $\mathbf{l}_i \times \mathbf{l}_j$ or $\mathbf{l}_j \times \mathbf{l}_i$. The terms $\mathbf{l}_i \times \mathbf{l}_j$ or $\mathbf{l}_j \times \mathbf{l}_i$ are two homogeneous points and negative to each other. They are equivalent in homogeneous coordinate. However, their corresponding points, on the normalized unit sphere, are opposite with regard to the origin. So those intersections, generating the same vanishing points, may converge into two clusters, opposite each other corresponding to the origin of the normalized unit sphere. In order to remove this malfunction, for any intersection homogenous point $\bar{\mathbf{x}} = [x \quad y \quad w]^{\mathrm{T}}$, we force $w > 0$. That is, if $w < 0$, we multiply -1 to the homogenous point. If $w = 0$, we force $y > 0$ such like what we do on w. If $y = 0$, we force $x > 0$. Through this rectification on all homogenous points of line intersections, we guarantee that all intersections, generating the same vanishing points, will converge into one cluster on the normalized unit sphere.

5. Experimental Results

The test will be implemented on synthetic data and real data. The synthetic data are with known vanishing points and with real imagery data. The synthetic data are generated under ideal projective transformation such that all parallel lines from the same direction intersect at a single vanishing point. The real data includes a field track picture and a brick wall picture. Because of the difference of fitted lines and true lines in real data, straight-line pairs intersect around the true vanishing point instead of hitting in the true vanishing point.

Test on synthetic data

The synthetic data are lines with three orthogonal directions. There are 20 lines in each direction, so there are $\binom{20}{2} = \frac{20\times 19}{2} = 190$ intersection pairs for each true vanishing point. Even in this ideal condition, the clustering algorithm must deal with the $3\times(20\times 20) = 1200$ cross points between pairs of lines with different directions.

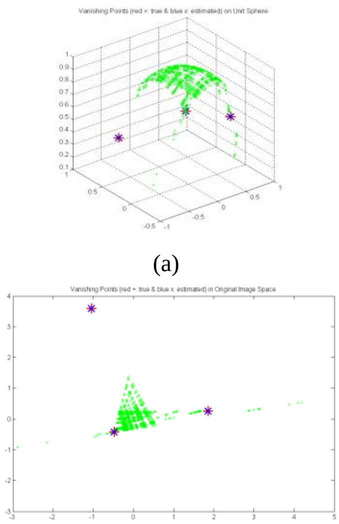
Figure 1 shows that the line intersection points on the normalized unit sphere and in the original image space for noise free synthetic data. Note that the parallel lines in one direction intersect at one point, the vanishing point in that direction. That is, there is a very strong cluster in each vanishing point. For noise free data, 190 intersections hit in one vanishing point in each direction. As a result, the K-means method can detect vanishing points very accurately. Table 1 and 2 show the true and estimated vanishing points and corresponding root mean square errors (RMSE). If the vanishing points are far from the cross points of lines, the estimated vanishing points equal the true ones. Or else, the estimated vanishing points agree well with the true ones.

Table 1. Vanishing Point Detection on Normalized Unit Sphere (for noise free synthetic data)

	TRUE	ESTIMATED	RMSE
VP 1	$\begin{bmatrix}0.8744\\0.1229\\0.4695\end{bmatrix}$	$\begin{bmatrix}0.8744\\0.1229\\0.4695\end{bmatrix}$	0
VP 2	$\begin{bmatrix}-0.2690\\0.9279\\0.2581\end{bmatrix}$	$\begin{bmatrix}-0.2690\\0.9279\\0.2581\end{bmatrix}$	0
VP 3	$\begin{bmatrix}-0.4039\\-0.3520\\0.8444\end{bmatrix}$	$\begin{bmatrix}-0.4041\\-0.3524\\0.8441\end{bmatrix}$	0.0005

Table 2. Vanishing Point Detection in Original Image Space (for noise free synthetic data)

	TRUE	ESTIMATED	RMSE
VP 1	$\begin{bmatrix}1.8624\\0.2617\\1.0000\end{bmatrix}$	$\begin{bmatrix}1.8624\\0.2617\\1.0000\end{bmatrix}$	0
VP 2	$\begin{bmatrix}-1.0421\\3.5944\\1.0000\end{bmatrix}$	$\begin{bmatrix}-1.0421\\3.5944\\1.0000\end{bmatrix}$	0
VP 3	$\begin{bmatrix}-0.4783\\-0.4169\\1.0000\end{bmatrix}$	$\begin{bmatrix}-0.4788\\-0.4175\\1.0000\end{bmatrix}$	0.0008



(a)

(b)

Figure 1. Vanishing point detection for noise free synthetic data. (a). on the normalized unit sphere (b). in the image space. The red '+' is the true position of vanishing point. The blue 'x' is the estimated position of vanishing point.

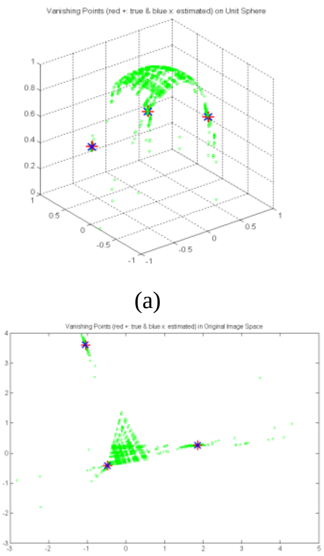
Figure 2 shows that the line intersection points on the normalized unit sphere and in the original image space for noisy synthetic data. Here, we add Gaussian noises to synthetic data. Note that the intersections of parallel lines in one direction (straight lines in image space) distribute around the vanishing point in that direction. There still exists a very strong cluster in each vanishing point. The K-means method can detect vanishing points well. Table 3 and 4 show the true and estimated vanishing points. We can see that the estimated vanishing points agree well with the true ones.

Table 3. Vanishing Point Detection on Normalized Unit Sphere (for noisy synthetic data)

	TRUE	ESTIMATED	RMSE
VP 1	$\begin{bmatrix}0.8744\\0.1229\\0.4695\end{bmatrix}$	$\begin{bmatrix}0.8742\\0.1227\\0.4697\end{bmatrix}$	0.0003
VP 2	$\begin{bmatrix}-0.2690\\0.9279\\0.2581\end{bmatrix}$	$\begin{bmatrix}-0.2692\\0.9281\\0.2572\end{bmatrix}$	0.0009
VP 3	$\begin{bmatrix}-0.4039\\-0.3520\\0.8444\end{bmatrix}$	$\begin{bmatrix}-0.4049\\-0.3530\\0.8434\end{bmatrix}$	0.0017

Table 4. Vanishing Point Detection in Original Image Space (for noisy synthetic data)

	TRUE	ESTIMATED	RMSE
VP 1	$\begin{bmatrix}1.8624\\0.2617\\1.0000\end{bmatrix}$	$\begin{bmatrix}1.8612\\0.2611\\1.0000\end{bmatrix}$	0.0013
VP 2	$\begin{bmatrix}-1.0421\\3.5944\\1.0000\end{bmatrix}$	$\begin{bmatrix}-1.0467\\3.6090\\1.0000\end{bmatrix}$	0.0153
VP 3	$\begin{bmatrix}-0.4783\\-0.4169\\1.0000\end{bmatrix}$	$\begin{bmatrix}-0.4801\\-0.4185\\1.0000\end{bmatrix}$	0.0024



(a)

(b)

Figure 2. Vanishing point detection for noisy synthetic data. (a). on the normalized unit sphere (b). in the image space. The red '+' is the true position of vanishing point. The blue 'x' is the estimated position of vanishing point.

Test on real data

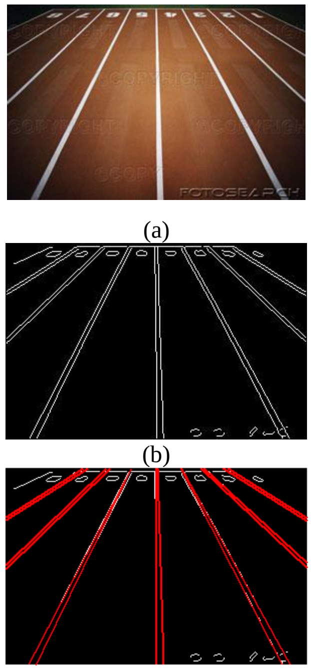
First, Canny edges are extracted on original images for further detecting straight lines. Then for each edge, we link continuous edge points into chains. The least square method is used to detect and fit straight lines on every linked edge point list. We define the average fitting error as the average perpendicular distance for each edge point to the fitted line. If the average fitting error is less than 0.4, we regard the edge as a straight line. In practice, we reject short lines. If its length is less than 20 pixels, we do not think that this line is meaningful. For the K-means method, the number of components is a difficult model selection problem, which is not considered here. In our experiments, we choose the number manually. Fortunately, under most cases for architectural buildings, there are only three clusters corresponding to three mutually orthogonal directions.

In order to compare with the estimates, we compute the true position of a vanishing point as the intersection of two parallel lines, where two points, manually selected, define each line.

Test on real data 1: the field track picture

In the field track picture, only the vertical parallel lines are clearly extracted using Canny edge detection, so the vanishing point exists in the vertical direction. We calculate edge linking on the Canny edges. Then the straight lines are fitted using the least square method on those linked points. Figure 3 shows the original picture, the detected Canny edges, and the fitted straight lines. Figure 4. shows the intersection points on the normalized unit sphere. We can see that the clustering property is still preserved. Therefore, We directly use Euclidean distance as similarity measure to cluster the points. There is only one cluster for the K-means. The detected vanishing points on the normalized unit sphere and in the original image space, agree well with the true vanishing points.

In this experiment, the number of fitted straight lines is 14. Hence, the number of intersection points is $\binom{14}{2} = \frac{14\times 13}{2} = 91$. The estimated position of the vanishing point on the normalized unit sphere is $[0.9554 \ \ -0.2952 \ \ 0.0064]^T$. The true position is $[0.9553 \ \ -0.2957 \ \ 0.0064]^T$. The estimated position of the vanishing point in the original image space is just the estimate in the normalized unit sphere divided by its third element, which is $[149.2813 \ \ -46.1250 \ \ 1.0000]^T$. The true position in the original image space is $[148.8293 \ \ -46.0701 \ \ 1.0000]^T$.

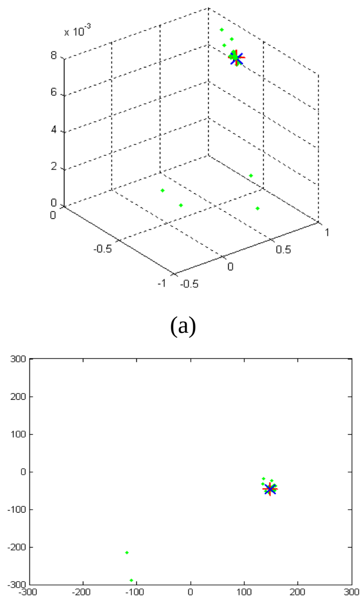


(a)

(b)

(c)

Figure 3. Test on field track picture. (a). original image (b). Canny edges (c). fitted straight lines (red lines) using least square method



(a)

(b)

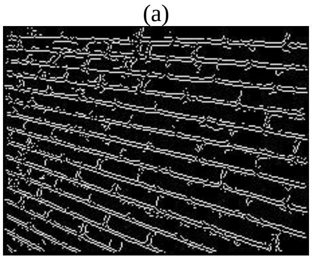
Figure 4. Vanishing point detection for field track picture. (a). on the normalized unit sphere (b). in the image space. The red '+' is the true position of vanishing point. The blue 'x' is the estimated position of vanishing point.

Test on real data 2: the brick wall data

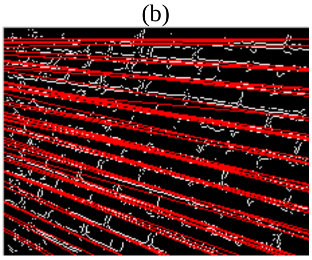
In the brick wall picture, only the horizontal parallel lines are clearly extracted using Canny edge detection. So we only estimate the vanishing point in the horizontal direction. We calculate edge linking on the Canny edges. Then the straight lines are fitted using the least square method on those linked points. Figure 5 shows the original picture, the detected Canny edges, and the fitted straight lines. Figure 6 shows that the line intersections on the normalized unit sphere and in the original image space. There is only one cluster for the K-means. Figure 6(a) shows that the detected vanishing point agrees well with the true vanishing point on the normalized unit sphere. Figure 6(b) shows the line intersection points in the original image space. The true vanishing point and the recovered estimated vanishing point from the unit sphere agree well.

In this experiment, the number of fitted straight lines is 59. Hence, the number of intersection points is $\binom{59}{2} = \frac{59 \times 58}{2} = 1711$. The estimated position of the vanishing point on the normalized unit sphere is $[-0.9972 \;\; 0.0024 \;\; 0.0037]^T$. The true position is $[-0.9999 \;\; -0.0154 \;\; 0.0036]^T$. The estimated position of the vanishing point in the original image space is just the estimate on the normalized unit sphere divided by its third element, which is $[-269.0146 \;\; 0.6340 \;\; 1.0000]^T$. The true position in the original image space is $[-279.1455 \;\; -4.2909 \;\; 1.0000]^T$.

(a)

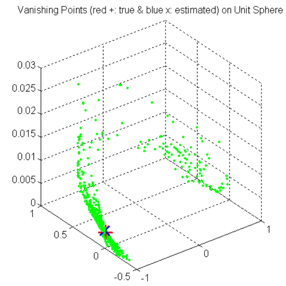


(b)

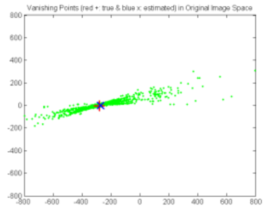


(c)

Figure 5. Test on brick wall picture. (a). original image (b). Canny edges (c). fitted straight lines (red lines) using least square method



(a)



(b)

Figure 6. Vanishing point detection for brick wall picture. (a). on the normalized unit sphere. (b). in the image space. The red '+' is the true position of vanishing point. The blue 'x' is the estimated position of vanishing point.

In comparison with the field track picture, we obtain much more accurate estimation results on the field track picture than those obtained from the brick wall picture. This is because we have much more accurate straight-line extraction on the field track image. Recall that the vanishing point is the intersection of straight lines.

The amazing property of the normalized unit sphere is that the clustering in the original image space is still preserved. In practice, the experimental accuracy depends on the edge and straight-line detection, straight line fitting, and clustering measure.

6. Conclusions

In this paper, we describe a general method for vanishing point detection and demonstrate its efficacy experimentally. In comparison with Gaussian sphere, our method transforms the non-homogeneous space into a normalized homogeneous space. In the normalized homogeneous space, finite points and points at infinity are treated equally.

From our results, we note that the transformation from the original infinite space into the limited space of a unit sphere can make a large-valued data to become small. Conversely, small estimate errors on the unit sphere may result in large estimate errors in the original space. Therefore, accurate edge detection and straight line extraction are very important. Nevertheless, the advantage of the normalized unit sphere is that the clustering in the original image space is preserved.

7. References

[1] S. T. Barnard. *Interpreting perspective images*. Artificial Intelligence 21, 1983, pages 435-462.

[2] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision* 4, 1990, pages 127–140.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge 2003.

[4] F. A. vanden Heuvel. *Vanishing point detection for architectural photogrammetr*y. International Archives of Photogrammetry and Remote Sensing, Vol. XXXII part 5, 1998, pages 652-659.

[5] D. Liebowitz and A. Zisserman. *Metric rectification for perspective images of planes*. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 1998, pages 482–488.

[6] E. Lutton, H. Maitre and J. Lopez-Krahe. *Contribution to the determination of vanishing points using Hough transfor*m. IEEE Transaction on Pattern Analysis and Machine Intelligence 16(4), 1994, pages 430-438.

[7] M. J. Magee and J. K. Aggarwal. *Determining Vanishing Points from Perspective Image*s. Computer Vision, Graphics and Image Processing 26, 1984, pages 256-267.

[8] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[9] L. Quan and R. Mohr. *Determining perspective structures using hierarchical Hough transform*. Pattern Recognition Letters 9, 1989, pages 279-286.

[10] C. Rother. *A new Approach for Vanishing Point Detection in Architectural Environments,* IVC 20(9-10), 2002, pages 647-656.

[11] M. Straforini, C. Coelho and M. Campani. *Extraction of vanishing points from images of indoor and outdoor scenes*. Image and Vision Computing, Vol. 11(2), 1993, pages 91-99.