Pattern Recognition
Manuscript Draft

Manuscript Number:

Title: Vanishing Hull: A Geometric Concept for Vanishing Points Detection and Analysis

Article Type: Full Length Article

Section/Category:

Keywords: Vanishing Point; Vanishing Hull; Edge Error Model

Manuscript Region of Origin:

Abstract: Vanishing points are valuable in many vision tasks such as orientation estimation, pose recovery and 3D reconstruction from a single image. This paper proposes a new concept from a geometric viewpoint, vanishing hull, which can be used to detect and quantitatively analyze the stability and accuracy of vanishing points. Given an edge error model, the range of a true edge can be modeled using a fan region. The geometric intersection of all these fan regions is a convex hull, which is called the vanishing hull. The vanishing hull gives the region of a true vanishing point, and its distribution determines the probability of the vanishing point. The expectation of the vanishing hull is the optimal solution of the vanishing point, its variance defines the accuracy of the estimation, and its shape determines the stability of the vanishing point. Hence, we can quantitatively analyze the stability and accuracy of the vanishing point estimation using the vanishing hull concept. Extensive simulation and real data results show that our method is significantly better than one state-of-the-art technique.

Vanishing Hull: A Geometric Concept for Vanishing Points Detection and Analysis

Jinhui Hu*, Suya You, Ulrich Neumann

University of Southern California, 3737Watt Way, PHE 404
Los Angeles, California 90089
Tel: 213-740-4250, Fax: 213-740-8931
{jinhuihu, suyay, uneumann}@graphics.usc.edu

Abstract

Vanishing points are valuable in many vision tasks such as orientation estimation, pose recovery and 3D reconstruction from a single image. This paper proposes a new concept from a geometric viewpoint, *vanishing hull*, which can be used to detect and quantitatively analyze the stability and accuracy of vanishing points. Given an edge error model, the range of a true edge can be modeled using a fan region. The geometric intersection of all these fan regions is a convex hull, which is called the vanishing hull. The vanishing hull gives the region of a true vanishing point, and its distribution determines the probability of the vanishing point. The expectation of the vanishing hull is the optimal solution of the vanishing point, its variance defines the accuracy of the estimation, and its shape determines the stability of the vanishing point. Hence, we can quantitatively analyze the stability and accuracy of the vanishing point estimation using the vanishing hull concept. Extensive simulation and real data results show that our method is significantly better than one state-of-the-art technique.

Keywords

Vanishing Point, Vanishing Hull, Edge Error Model

* Corresponding author. Email: jinhuihu@graphics.usc.edu. Tel: 1-213-740-4250, Fax: 1-213-740-8931.

1 Introduction

A vanishing point is defined as the intersection point of a group of image lines that correspond to the projection of parallel lines in 3D with an ideal pin-hole camera model. The position of a vanishing point in the image plane is only determined by the camera center and the orientation of the 3D lines in the camera system. Vanishing points are valuable in many vision tasks. A traditional application of vanishing points is building detection in aerial images [1]. Vanishing points can be used to group images lines, which are then used to form hypotheses of building edges. Cameras can be calibrated using three orthogonal vanishing points [2,3], and pose can be recovered [4,5]. Other applications of vanishing points include robot navigation [6] and 3D reconstructing from a single image [7,8,9]. The goal of this paper is to present a new consistent framework for vanishing points detection, and stability and accuracy analysis. We present extensions and new capabilities, and discuss theory and algorithm details that were omitted or summarized in our previous work [10].

1.1 Motivation

Much research has been conducted on accurately identifying the position of vanishing points. This varies from simple line grouping [11] to more complicated methods using statistical models [12]. Most previous research work focuses on finding the group of lines corresponding to valid vanishing points, or the vanishing points detection problem, and the performances are often evaluated empirically [1]. However, there is very little research focused on finding a theory to quantitatively analyze the stability and accuracy of vanishing points estimation. This is the main motivation of this work.

Different from most previous work [13-18], we attack this problem from a geometric viewpoint. Observing that the region spanned by lines sampled from an edge error model is a fan region (Figure 1, Section 2.1), we intersect all the fan regions to form a convex polygon, called the *Vanishing Hull*. This papers shows that the vanishing hull has some interesting properties and it can lead us to an optimal estimation of vanishing points. In more details, the vanishing hull gives the region of the true vanishing point, and its distribution gives the probability of the vanishing point. The expectation of the vanishing hull, the centroid for a uniform distribution, gives the optimal solution of the vanishing point under statistical meaning, its variance defines the accuracy of the vanishing point and its shape determines the stability of

the vanishing point. Hence, the vanishing hull concept provides a theoretical framework to quantitatively analyze the region, optimal solution, stability and accuracy of vanishing points. Besides a framework for analyzing vanishing points with image noises, we also present a novel geometric edge grouping method based on edge error models.

Before continuing, we would like to point out that we are not the first one to explore the idea of estimating the probability distribution of geometric entities in compute vision. Similar work has been done by Wolfgan Forstner and Stephan Heuel[19,20]. Forstner and Heuel combine geometry and statistics uncertainty [21-23] for line grouping [20] and 3D reconstruction [19,24]. While their work convert the geometric problem of joint and union into an algebra problem using double algebra [25-28], we are more focused on the geometric entities, such as the shape and centroid of the vanishing hull. More significantly, we avoid the usage of covariance propagation, which becomes cumbersome when the number of lines are large, a typical situation for vanishing points estimation. Forstner and Heuel's method is suitable for 3D reconstruction with pairs of lines, while our method is more suitable for vanishing points estimation with large number of lines[1].

1.2 Related work

There are two key problems in identifying vanishing points, finding the group of image lines that correspond to a true vanishing point and computing the position of the vanishing point with the presence of image noises. According to the methodology used in the two steps, we classify different methods into two classes, clustering methods and voting methods.

Clustering methods first find possible clusters using the intersection of all pairs of lines [29] or image gradient orientations [30]. Then a criterion of distance or angle is used to assign each line to different clusters. The drawbacks of clustering methods are the high computational complexity and that a hard threshold is needed to group lines into clusters. Liebowitz and Zisserman [29] first group the image lines, then use histogram to find the dominant vanishing points. A Maximum-Likelyhood estimator is used to compute the position of the vanishing point. McLena and Kotturi [30] integrate edge detection and line clustering to the process of vanishing points detection, and then use a non-linear method to compute the

[1] Our experiments show that the larger the number of lines, the more accurate the vanishing point estimation.

position of vanishing points with a statistical edge error model.

Voting methods can be classified into image space methods and Gaussian sphere methods according to the space they use for voting. Rother[31] accumulates the votes of image lines in the image space, then search the vanishing points using knowledge such as orthogonal criteria and camera criteria. However, this method is computational expensive.

A popular method is using the Hough Transform in Gaussian sphere space [32], which is a global feature extraction method [2]. Many improvements have been made to address the shortcomings of Hough-based approaches[12,33]. Shufelt [1] uses the knowledge of primitive models to reduce spurious maxima, and an edge error model to improve the robustness to image noises. The drawback of voting in Gaussian sphere is that the accuracy is limited to the discretization of the accumulator space, hence it is hard to achieve the precision that the image can provide. Antone and Teller [4,5,15] use a hybrid method of Hough Transform and least square to detect and compute vanishing points. G. Schindler and F. Dellaert [34] use a EM method to compute vanishing points in *Manhattan world* [35].

Recently, Almansa et al. [13] propose a system using vanishing regions of equal probability to detect vanishing points without any a priori information. A vanishing region is different from a vanishing hull, the former is an accumulation space for vanishing points detection while the latter is a tool to analyze the stability and accuracy of vanishing points.

The rest of the paper is organized as following. We first introduce the vanishing hull concept and its properties based on a simple end-points edge error model (Section 2), then we present some novel methods for vanishing points detection based on the edge error model (Section 3). Section 4 presents an algorithm to determine the vanishing hull and analyze vanishing points estimation, and we extend the vanishing hull concept to general edge error models in Section 5. The performance of our method is extensively analyzed with both simulation and real data, and quantitatively compared with one state-of-the-art technique [29](Section 6). Finally, we present some applications in Section 7 and conclude the paper in Section 8.

2 Vanishing Hull

Since the idea of vanishing hull is derived from the intersection of edge regions, we first present a simple edge error model, then introduce the definition and properties of the vanishing hull.

2.1 Edge error model

Various edge error models have been presented. McLean and Kotturi [30] use a statistical model to present both the error of the line centroid and the error of its orientation. Other models using both geometry and statistics can be found in [19]. Shufelt [1] presents a simple but effective edge error model. Inspired by the idea of a fan edge region, we derive the concept of vanishing hull by the intersection of all these regions. We first adapt this simple edge error model to our vanishing hull framework, and then extend the concept to general edge error models (Section 5).

Consider the representation of a line segment using two endpoints, and assume that the two end points have one pixel precision, then two fan regions with a rectangular region in the middle can be formed by moving the two end points freely in the pixel squares (Figure 1). This region is a concave region, so we cannot guarantee that the intersection of such regions will be convex. Fortunately, a true vanishing point cannot lie in the image line segment (Section 3.1), so the rectangular region has no effect on the shape of the intersection of edge regions. We simply take the middle point of the edge, and form two fan regions with the two end points. Furthermore, a true vanishing point can only lie in one direction of the edge, so we just take one of the fan region, which is a convex region (Figure 1).

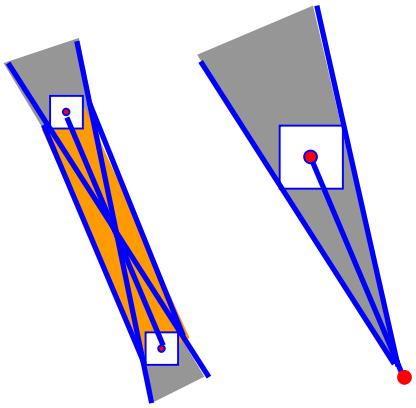


Figure 1. Edge error model.

2.2 Vanishing hull definition and property

The *Vanishing Hull* is defined as the intersection of the fan-shape edge regions with a given edge error model (Figure 2). Figure 15 shows a vanishing hull of a real image. We first present the properties of the vanishing hull, the computation and algorithm details are presented in section 4.

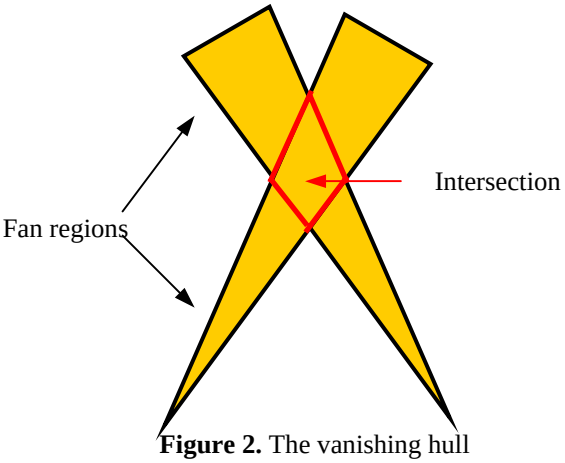


Figure 2. The vanishing hull

Property I. A vanishing hull is not empty. Proof: according to the grouping method that will be presented

shortly in Section 3.1, an edge is assigned to a cluster only if its edge region covers the intersection point of the cluster, so the vanishing hull contains at least the intersection point.

Property II. A vanishing hull is convex. Proof: the intersection of convex regions (edge regions are fan shape regions, hence convex) is convex.

Property III. A true vanishing point lies inside the region of its vanishing hull with the assumption that the edge error model is correct (the true edge lies inside the fan region). Proof: By definition, the true vanishing point must lies inside the union of all the edge regions. Now assume the vanishing point *VP* lies outside of the vanishing hull but inside the union of the edge region. Then there must exist some edge, say *L*, whose edge region does not cover *VP*. Hence the edge error model of edge *L* is wrong, which is contradictory with our assumption. Hence, *VP* must be inside the vanishing hull. This property is important, it tells us where to find the true vanishing point.

Property IV. The centroid of a vanishing hull is the optimal estimation of the vanishing point with a uniform distribution model. Proof: the optimal estimation of the vanishing point is the expectation of the probability distribution of the vanishing points inside the vanishing hull under statistical meaning. With a uniform distribution, the expectation of a vanishing hull is its centroid.

Property V. The variance of a vanishing hull determines the accuracy of the estimated vanishing point. Proof: this follows directly from the probability theory.

Property VI. The shape of a vanishing hull determines the stability of the estimation of the vanishing point.

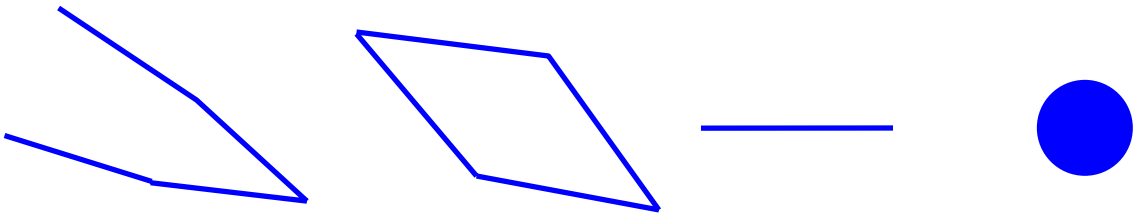


Figure 3. The shape of a vanishing hull. From left to right, open, close, a line and a point.

A vanish hull can be open, it can also be a closed non-trivial convex polygon, a line segment or a point (Figure 3). When the image lines are parallel, the vanishing hull is an open convex hull (Figure 3 (a)), the centroid is undetermined, which means the estimation of the vanishing point is unstable. This is reasonable because edges have noises, any non-zero noise will be enlarged to infinity when the vanishing point is at

infinity, which makes the estimation unreliable. An open vanishing hull indicates a vanishing point at infinity, which corresponds to a point at the great circle parallel to the image plane in the Gaussian sphere. We handle the open vanishing hull case by setting the vanishing point to infinity.

When the vanishing hull is a closed non-trivial convex polygon (Figure 3 (b)), the vanishing point can be estimated using the centroid with the variance of the distribution as the estimation accuracy. When the vanishing hull shrinks to a line segment (Figure 3 (c)), the uncertainty is along just one direction, and the vanishing point can be precisely computed when the vanishing hull degenerates to a point (Figure 3 (d)), which corresponds to an error-free edge model.

3 Vanishing Points Detection

Since we are interested in finding the intersection regions of all the edges, it is necessary to identify the image lines that can form a possible vanishing point, i.e., we need to group lines into different clusters to detect vanishing points. We describe the grouping process in both image space and the Gaussian sphere.

3.1 Image space

Finding clusters

The Canny edge detector is used to find sub-pixel edges, then Hough Transform is used to find possible lines, and nearby lines are connected using some user defined threshold. The lines are represented using two end points.

The intersections of all pairs of line segments are computed to find all possible clusters. The computational complexity is $O(n^2)$, where n is the number of lines. Grouping lines into different clusters takes O(n) time, so the overall time complexity is $O(n^3)$, which is expensive for a large number of lines. We will reduce the complexity using a filtering step and the RANSAC algorithm later.

Grouping

After finding the clusters, we need a criterion to assign lines to different clusters. A distance criterion gives priority to close vanishing points, while an angle criterion gives priority to far vanishing points. A reasonable threshold is to use a tuple of both distance and angle, or use normalized angle error [30]. However, all these methods need a hard threshold, which may be inconsistent with the edge error model.

We use a geometric grouping method that is consistent with the edge error model without any hard thresholds. For each cluster of two lines, we can find the intersection region *A* of the edges, and a test edge is assigned to this cluster when its edge region overlaps with region *A*. Furthermore, we use a strong constraint for clustering (Figure 4). An edge is assigned to a cluster only if its edge region covers the intersection point of the cluster. This guarantees that the intersection region of the edge regions in each cluster is not empty (Property I in Section 2.2). The normalized length of each edge is accumulated in its assigned cluster, and the maximum clusters are chosen to compute potential vanishing points.

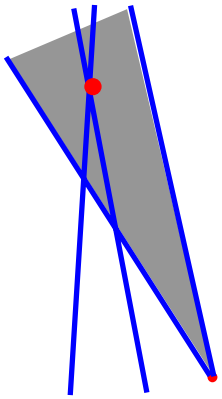


Figure 4. Grouping with an edge error model.

Filtering spurious vanishing points

Most of our testing images are outdoor building images with heavy occlusion by trees (Figure 9 (a)), which causes many spurious vanishing points. Knowledge of the image and vanishing points are used to filter spurious vanishing points. First we roughly classify the extracted lines into *x* and *y* groups[2] according to the line orientation to reduce the size of line number. Then vanishing points are filtered using the following three filters.

1) Iterative line length. According to the edge error model, longer lines are more reliable, however, we would like also to keep shorter lines. So we first filter the lines using a large length threshold, then estimate the possible vanishing points, and these points are used to find more short line supporters according to the grouping method.

2) Covering area. Another observation of the image is that edges of trees only cover a small part of the image region, so the ratio of the covering area against the image area is also used to filter spurious vanishing points.

3) Valid vanishing point. Vanishing points are the intersection of image lines that correspond to parallel lines in 3D. So by definition, a valid vanishing point will not lie on the image segment in the image space. This filter is very effective in reducing spurious clusters.

[2] Each group may contain more than one vanishing point.

RANSAC

Even though we classify lines into two groups to reduce the line number, and use filters to reject spurious clusters, the number of clusters may still be large. Since we are interested in find vanishing points that correspond to dominant directions, the RANSAC algorithm is used to find the maximum cluster of lines. Generally there exist three orthogonal vanishing points for images of outdoor buildings. We first find the dominant vanishing points for *x* and *y* direction. The vanishing point of z direction is estimated using the orthogonal property of the three directions, and its supporting lines are found using our grouping method to refine the position using the vanishing hull.

3.2 Gaussian sphere

The geometric grouping method can also be adapted to the Gaussian sphere space. According to the edge error model, each line casts a swath on the Gaussian sphere rather than a great circle. Similar to Shufelt [1], we can rasterize the swath into the Gaussian sphere using polygon boundary fill algorithm. The three heuristic filters can be adapted as well. For the iterative line length filter, we first only scan lines with length above some threshold into the Gaussian sphere, then discretize the sphere and collect votes weighted with line length. Then we find the dominant cell and scan more short lines to find more supporting lines. For the other two filters, we find the corresponding lines for each dominant cell, and compute the covering area and filter them with a threshold, and finally test whether the dominant cells lie out of each line segment.

The Gaussian sphere method has the advantage of treating each vanishing point equally, including infinite vanishing point. However, in experiments with real images, we opt to use the clustering method in image space rather than the Gaussian sphere for several reasons. First, the accuracy of the Gaussian sphere is limited to the discretization accuracy, hence it is hard to achieve the precision that an image can offer. Secondly, the intersection of the fan regions of the edges that belong to the maximum cell in the Gaussian sphere may be empty, which makes the vanishing hull meaningless. While in the image space, we can guarantee the vanishing hull to be non-empty using the aforementioned grouping method. The last reason is that the projected vanishing hull onto the Gaussian sphere is not a polygon anymore, so it is hard to compute the probability distribution and analyze the stability and accuracy. Furthermore, we argue that we treat finite and infinite vanishing points equally even using image space grouping method. This is due to

Property VI of the vanishing hull. When the vanishing point is finite, it can be well determined by the vanishing hull. When the vanishing point is at infinity, the vanishing hull is open, which can be easily detected. We set the orientation angle for an infinite vanishing point to zero.

4. Determining Vanishing Hull

4.1 Half-plane intersection

Given the group of lines that form a vanishing point, let us consider how to find the vanishing hull. A fan shape edge region can be considered as the intersection of two half planes (Figure 1), so the problem of finding the intersection of the edge regions can be cast as the problem of half-planes intersection. A naïve way to solve the problem is to add one half-plane bound line at a time to compute the intersection region, which takes $O(n^2)$ time. However, a more elegant algorithm with $O(n \lg n)$ time can be presented by utilizing the properties of dual space [36].

There exists an interesting property called duality between lines and points [36]. Given a non-vertical line L, it can be expressed using two parameters (k, b). We can define a point with k as the x coordinate and b as the y coordinate. This point is called the dual point (L^*) of line L. The space of the lines is called the prime space, while the space of the points is called the dual space. The half-plane intersection problem in the prime space can be mapped as the problem of finding the convex hull of the points in the dual space.

We first divide the bound lines of half-planes into two sets: an upper set (half-planes lie above the bound lines) and a lower set (half-planes lie below the bound lines). Let's consider the upper set first. For each line L in the upper set, we can find a dual point $L^*(k, b)$. The intersection of all the half-planes in the upper set can be mapped as finding the lower convex hull (the edges of the lower boundary of the convex hull) of the dual points. The proof is out of the scope of this paper. Readers may refer to [36]. Similarly, we can find the intersection of all the half-planes in the lower set by determining the upper convex hull of the dual points. Note that according to our dual mapping, the upper convex hull and the lower convex hull will not intersect although the upper and lower half-planes do intersect, and that is the reason we split the half planes into two sets. Finally, the two regions are merged to find the vanishing hull. The algorithm is summarized as following:

1. *Split the half plane into two sets, an upper set and a lower set.*
2. *Map each set into a dual space and find the corresponding upper and lower convex hull (note that the two convex hulls are different).*
3. *Map the two half convex hulls back to prime space, and merge them to find the vanishing hull.*

Finding a convex hull of a point sets is a well-defined problem [36], which takes $O(n\lg n)$, with n as the number of points. The mapping and merging takes linear time, so the overall time complexity is $O(n\lg n)$.

4.2 Expectation and variance of vanishing hull

As aforementioned, we can estimate the position of the vanishing point using the expectation of the vanishing hull and analyze the accuracy using its variance. Now let us consider the computation of the expectation and variance of the vanishing hull. With a uniform probability distribution, the expectation and variance of the position of a true vanishing point can be computed using Equation 1, where *D* is the region of the vanishing hull, and *A* is its area. We only show *x* coordinate here, *y* coordinate can be computed in a similar way. Given a list of vertices of the vanishing hull, it is easy to show that the mean can be computed using the coordinates of these vertices (Equation 2, 3).

$$\mu(x) = \frac{1}{A}\iint_D x dA \qquad \mathrm{var}(x) = \frac{1}{A}\iint_D (x-\mu(x))^2 dA \tag{1}$$

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \tag{2}$$

$$\mu(x) = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{3}$$

The variance of the vanishing hull can also be represented as a simple expression of the coordinates of the vertices. According to Green's theorem [37], an integral in the region can be converted to an integral on the boundary:

$$\int_{\partial D} f(x,y)dx + g(x,y)dy = \iint_D (\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y})dxdy \tag{4}$$

Let $f = 1, \quad g = \frac{1}{3}(x-\mu(x))^3$, we have

$$\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y} = (x-\mu(x))^2 \tag{5}$$

So: $\text{var}(x) = \frac{1}{A}\iint_D (x-\mu(x))^2\,dA = \frac{1}{A}\iint_D (\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y})dA$

$$= \frac{1}{A}\int_{\partial D} f(x,y)dx + g(x,y)dy \tag{6}$$

Let us consider the integral on line (x_i, y_i) to (x_{i+1}, y_{i+1}), we can parameterize the point on the line using *t*, such that:

$$x = x_i + t(x_{i+1} - x_{i)} \qquad y = y_i + t(y_{i+1} - y_{i)} \quad t \in [0,1] \tag{7}$$

Let $a_i = (x_i - \mu(x))$, $b_i = (x_{i+1} - x_i)$, $c_i = (y_{i+1} - y_i)$, it is easy to show that the variance of *x* coordinate can be computed as:

$$\text{var}(x) = \frac{1}{3}\sum_{i=0}^{n-1} c_i (a_i{}^3 + \frac{3}{2} a_i{}^2 b_i + a_i b_i{}^2 + \frac{b_i{}^3}{4}) \tag{8}$$

Similarly, we can compute the variance of *y* coordinate.

5 Vanishing Hull for General Edge Error Model

The vanishing hull concept can be easily extended to general statistical edge error models. First we show an augmented vanishing hull considering the full edge region of the simple edge error model, then we show that a general vanishing hull can be derived in a similar way.

In section 2.1 we ignored the rectangular region, and used one of the edge fan to derive the vanishing hull concept. We claimed that the shape of the vanishing hull will not change with this approximation, which is true. However, the probability distribution in the vanishing hull is not a uniform distribution. For a full edge span region, an angle (called extreme angle θ) is formed by the vanishing point *VP* and two extreme points P_1 and P_2 (Figure 5). According to the edge error model, the two end points have equal probability inside the one-pixel-size square. The probability of a true edge passing through the vanishing point *VP* is determined by the overlapping area of the extreme angle and the pixel squares.

Let $p_i(l_i, VP)$ be the probability of a true edge l_i passing through *VP*, e_1 and e_2 be the two end points of l_i, and S_1 and S_2 be the overlapping region of the extreme angle with the two pixel squares, then:

$$p_i(l_i, VP) = p(e_1 \in S_1 \,\&\, e_2 \in S_2) \tag{9}$$

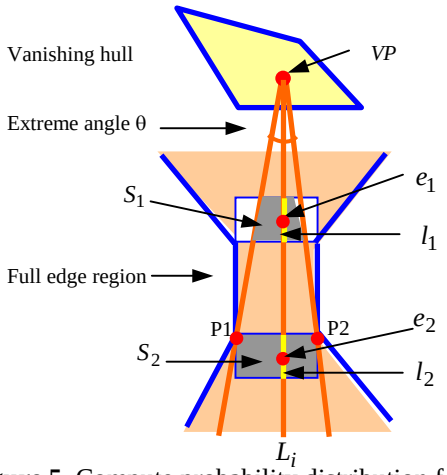


Figure 5. Compute probability distribution for a full edge region.

Assuming the two end points are independent with probability density function (PDF) $f(x,y)$ and $g(x',y')$ respectively, then the joint PDF is $f(x,y)*g(x',y')$, which is a 4D uniform distribution. Then $p_i(l_i,VP)$ is the integral of the joint PDF over the region $S_1\ S_2$.

$$p_i(l_i,VP) = \iint_{e_1\in S_1 \& e_2\in S_2} f(x,y)*g(x',y')dS_1dS_2 \tag{10}$$

Note that points e_1, e_2 and VP are collinear. We can use a line-sweeping method to compute the integral in a region. Now consider a sweep line L_i that passes VP (Figure 5), and intersects the two squares with line segment l_1 and l_2 (Note that L_i is a line inside the extreme angle, while l_i is a true edge inside the whole edge region). Line L_i sweeps the whole overlapping region when its angle varies from 0 to θ, the integral of the joint PDF over the two line segments is:

$$p(e_1\in l_1 \,\&\, e_2\in l_2) = \int_{e_1\in l_1 \& e_2\in l_2} f(x,y)*g(x',y')dl_1dl_2 \tag{11}$$

Denote the end points of line segment l_1 as $(x_1,y_1)(x_2,y_2)$ and l_2 as $(x_1',y_1')(x_2',y_2')$, they form a line segment in a 4D space (because we have three linear constraints: l_1, l_2 and they have the same slope). The integral of a uniform distribution over a line segment in 4D is just the length of the line segment, hence:

$$p(e_1\in l_1 \,\&\, e_2\in l_2) = [(x_2-x_1)^2+(y_2-y_1)^2 + (x_2'-x_1')^2+(y_2'-y_1')^2]^{\frac{1}{2}} \tag{12}$$

Now we can integrate over the angle to get the region integral:

$$p_i(l_i, VP) = \int_{\vartheta} p(e_1 \in l_1(\vartheta) \,\&\, e_2 \in l_2(\vartheta)) d\vartheta \tag{13}$$

The full expression of the analytical PDF of the vanishing hull for a full edge region is complicated, and the distribution function may not be continuous over the entire region. Even when the distribution for a single edge is continuous, the overall PDF is very high order due to the large number of edges. An analytical solution to the integral of such a high order PDF is very complicated, and may not exist. We use a discretizing method to solve this problem. The discretization process can achieve high precision (one pixel) because the vanishing hull region is bounded. Consider the center point, $VP(x, y)$, of a cell with one pixel size, we can find the extreme angle relative to edge l_i, and then compute the probability $p_i(l_i, VP(x, y))$ according to Equation 13. Then the probability of the point $VP(x, y)$ over all the edges is computed and normalized (Equation 14). The expectation and variance can be easily computed in the discretized vanishing hull.

$$P(x, y) = \prod_{i=0}^{n-1} p_i(l_i, VP(x, y)) \qquad P^*(x, y) = \frac{P(x, y)}{\sum_D P(x, y)} \tag{14}$$

Such a vanishing hull considering the full edge span region is called “augmented vanishing hull”. In practice, we found that the vanishing hull often consists of only a few vertices (less than 10 vertices for 1000 lines), which means that the probability of a vanishing point being close to the edge region’s boundary is very low. Since vanishing points close to the middle of the edge region has similar overlapping area with the pixel squares, it is reasonable to assume a uniform distribution for the vanishing hull formed of full edge regions.

We can extend the vanishing hull concept to general edge error model in a way similar to the augmented vanishing hull. A general edge error model often models the error of the edge centroid and orientation, the two end points or edge pixels, using a Gaussian distribution. The edge span region is still a fan shape, so the intersection of the edge regions is a convex hull. Assuming the PDF of the line l_i passing a vanishing point (x, y) is $f_i(x, y)$, then the PDF of the vanishing hull over all the lines is $\prod_i f_i(x, y)$. Again, this function is a high order non-linear function, we discretize the vanishing hull, and compute the mean and variance of the vanishing hull.

6 Analysis and Results

We have extensively analyzed the vanishing hull concept using both synthetic and real data. The performance of our method is also compared to one state-of-the-art algorithm [29].

6.1 Simulation data

We first analyze the vanishing hull concept using synthetic data. The goal of the simulation is to show that a vanishing hull gives the region of the true vanishing point, its expectation is the optimal solution, its shape determines the stability and its variance determines the accuracy of the estimation. The simulation is designed as following. A group of 3D parallel lines are projected by an ideal pin-hole camera to an image plane, then random noises with specified magnitude are added to the end points. The vanishing point is estimated using the centroid of the vanishing hull assuming a uniform distribution. We extensively analyze our Vanishing Hull (VH) algorithm with different parameter settings (Table 1). For each of the five groups of parameter settings, we sample the space with 100 evenly distributed intervals, the other four parameters are set as constant when one parameter varies in its range to test the performance relative to each parameter.

Table 1. Parameter settings.

Parameter	Range	Other parameter settings
1.line orientation angle (degree)	$\theta \in [0.01, 40]$	$fov = 40$ $l = 50$ $\varepsilon = 0.5$ $n = 200$
2.camera field of view (degree)	$fov \in [20, 80]$	$\theta = 1$ $l = 50$ $\varepsilon = 0.5$ $n = 200$
3.image line length (pixel)	$l \in [10, 100]$	$\theta = 10$ $fov = 40$ $\varepsilon = 0.5$ $n = 200$
4.image noise magnitude (pixel)	$\varepsilon \in [0.05, 0.5]$	$\theta = 1$ $fov = 40$ $l = 50$ $n = 200$
5.number of image lines	$n \in [20, 1000]$	$\theta = 5$ $fov = 40$ $l = 50$ $\varepsilon = 0.5$

Vanishing hull is the true vanishing point region

The simulation shows that all the true vanishing points lie inside the vanishing hull. This is logical because the maximum noise magnitude is specified, so the edge error model exactly predicts the region of the true edge region, hence the true vanishing points lie inside the vanishing hull.

Expectation is the optimal solution

The result of VH method is compared with other two methods, Least Square (LS) method and Maximum Likelyhood (ML) method [29], to show that the expectation is the optimal solution. The comparison criterion is the recovered orientation angle error relative to the ground truth. LS method uses least square to find the vanishing point closest to all lines, and ML method uses a non-linear method to minimize the distance of the line that passes the vanishing point and the mid-point to the two end points. According to our implementation, the difference of the ML and LS method is often several pixels, so the angle difference is very small. This is because ML method uses a non-linear optimization method, which often gives a local minimum close to the result of LS method. We just show the result of VH and ML method to make the figure clear (Figure 6).

The parameter θ and fov are related to the perspective effect of images. The result (Figure 6 (a), (b)) shows that the ML method gives large errors for weak perspective images (up to 40 degrees when the orientation angle is less than 0.1 degrees), while VH method performs reliably with maximum angle error less than 0.5 degrees and average less than 0.1 degrees. When the perspective effect is strong (orientation angle is larger than 10 degrees), the vanishing hull shrinks to several dozens of pixels, both methods perform well.

The parameter l and ε are related to the quality of edges. The simulation (Figure 6 (c), (d)) shows that the ML method gives large orientation errors for poor quality edges ($l < 30$ or $\varepsilon > 0.2$), and the maximum angle error is more than 14 degrees. The VH method performs significantly better than the ML method and very reliably over the whole range (maximum angle error 0.3 degrees, and average angle error less than 0.1 degrees).

The last parameter (Figure 6 (e)) compares the performance against the number of image lines. The simulation shows that the number of lines has no strong effects on the performance of ML method, however, it affects the performance of the VH method. When the number is more than 500, VH method performs very reliably, when the number drops to 100, it still performs significantly better than the ML method. However, when the number of lines drops below 50, the result is mixed. There are several reasons for this. First, the region of a vanishing hull shrinks with the increasing number of lines, so it is more reliable for more lines. Second, the expectation of the vanishing hull is the optimal solution for a vanishing point under statistical meaning. However, when the sample (number of lines) is small, the true value of the

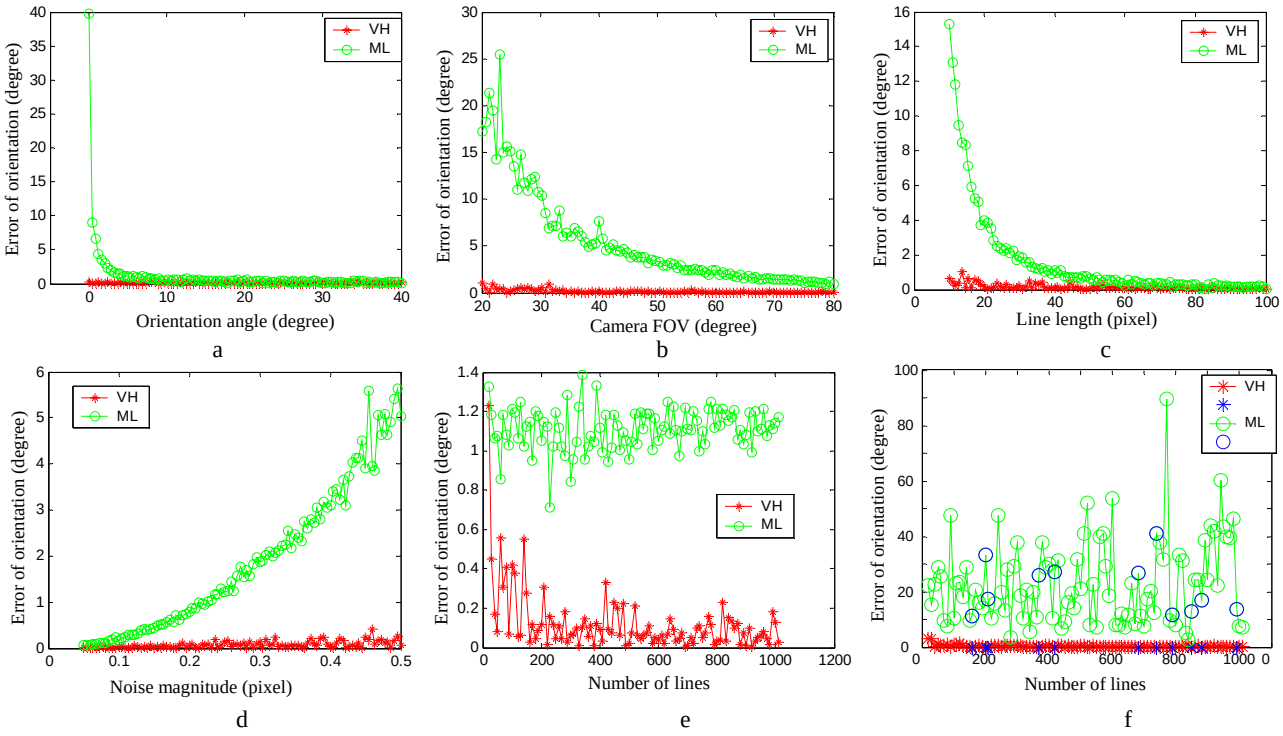


Figure 6. Performance comparison of VH and ML method on synthetic data with random noise.

vanishing point may deviate from the statistical value. The last reason is that we use a uniform distribution, which is an approximation as we showed in section 5.

In general, the VH method performs significantly better than the other two methods, especially for weak perspective images, and the performance of the VH method is very reliable with several hundreds of lines, a reasonable number for high-resolution images of buildings and aerial images. This shows that the expectation of the vanishing hull is the optimal solution of vanishing points estimation.

Stability and accuracy

We can also predict unstable vanishing points using the VH method. For most of the cases, the vanishing hull is closed, which indicates that the vanishing point is stable. When the vanishing hull is open, it indicates that the vanishing point is at infinity. The algorithm simply sets the orientation angle to zero for infinite vanishing points. We visualize unstable vanishing points with blue color in Figure 6 (f), where the ground truth $\theta = 0.01$. By setting the orientation of unstable vanishing points to zero, the VH method has a small error of 0.01 degree, while the ML method gives a large error (more than 10 degrees). Note that the VH method achieves an average error less than 0.5 degrees even for such ill-conditioned cases. The error of

the VH method is also within the magnitude of the variance for all five groups of parameters, which implies that the variance determines the accuracy of the estimation. The graph of the variance is not shown here.

Since the noises in real images are generally Gaussian noises, we have also tested the performance of our algorithm with the Gaussian noises. Similar to the random noise case, we project a group of 3D parallel lines with an ideal pin-hole camera to an image plane, then add Gaussian noises with zero mean and specified variance to the end points. Again, we test the performance with five groups of parameter settings (Table 1). The vanishing point is estimated using the centroid of the vanishing hull rather than discretizing the vanishing hull and computing the exact distribution. It is interesting to note that, although the distribution is not uniform in the vanishing hull with the Gaussian noises, the centroid is still a very good approximation to the optimal solution. The reason might be that the optimal solution is close to the centroid with zero mean Gaussian noises. Of course, one can also compute the exact optimal solution using Equation 14, however, we find that the centroid approximation is good enough and much simpler to compute. We also compared the performance of VH method with the ML method (Figure 7), the result is similar to that of the random noises.

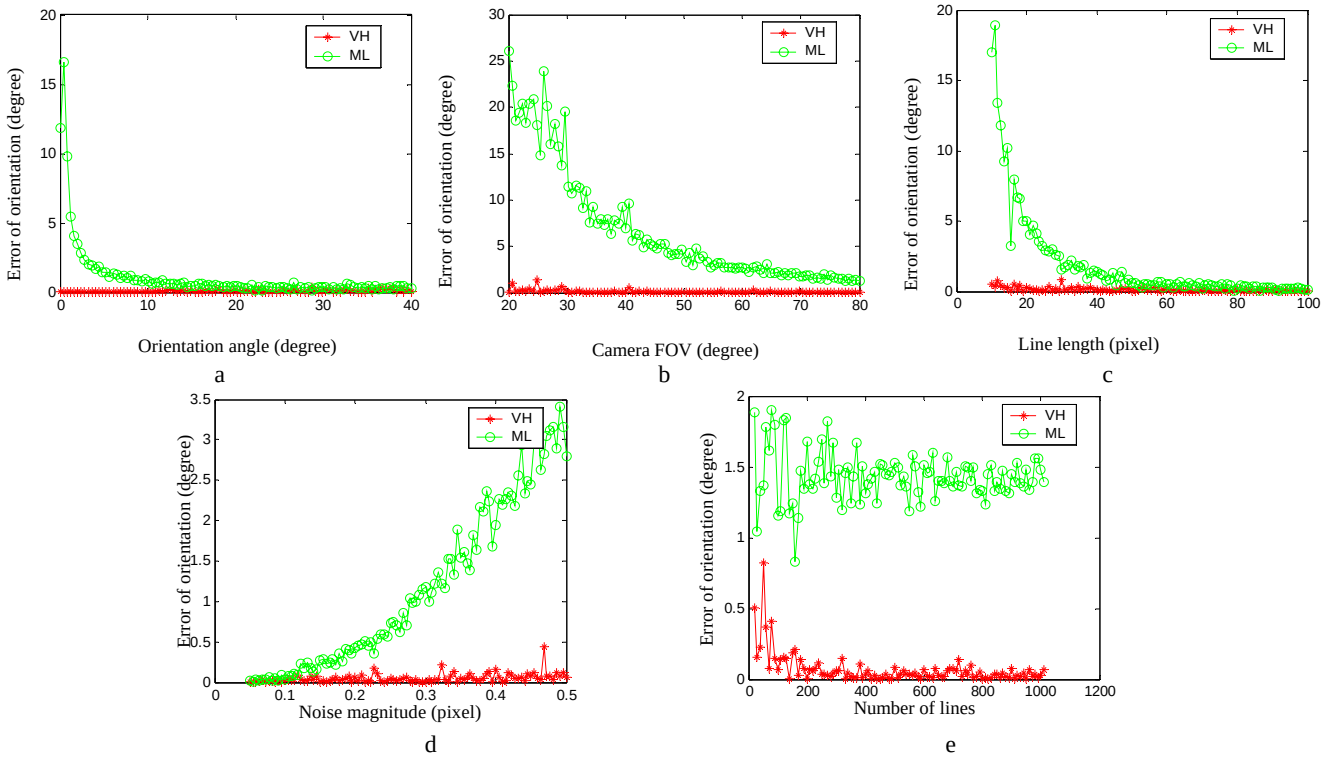


Figure 7. Performance comparison of VH and ML method on synthetic data with Gaussian noise.

6.2 Real data

The vanishing hull concept is also extensively tested with real images to generate textures (Section 7). However, the comparison of the performances of different algorithms in computing vanishing points for real images is difficult. The main challenge is that the ground truth positions of vanishing points for real images are unknown. Furthermore, the errors may come from different sources, such as the error of edge model, grouping errors, camera lens distortion and camera calibration errors. Here we compare the performance of the VH and ML method with two sets of images, indoor and outdoor images. More applications of the VH method for real images are presented in section 7.

Indoor images

For the indoor case, we print some pattern of parallel lines, and carefully place the camera with a tripod so that the image plane is parallel to the pattern plane. We translate the camera to take 15 different pictures while keeping the orientation unchanged. Figure 8 shows one of such images. By carefully controlling the camera motion, we can assume that the orientation ground truth is close to zero (careful user verification using the images shows that the variance is within 0.5 degrees). We have also tried to control the camera motion to take images with a slanted angle, however, it is very hard to measure the exact orientation angle. Furthermore, as shown in the graph of Figure 6 (a), when the rotation angle is more than 10 degrees, although the VH method still performs better than the ML method, the difference is so small that it will be covered by the measurement error.

To reduce the error from the camera, we calibrate the camera and correct the lens distortion for all the 15 images. As aforementioned, the vanishing hull concept is derived from the edge error model, so its performance depends on the edge error model. For real images, we model the noise magnitude of the end points of edges using an empirical model: $\varepsilon = c/\sqrt{l}$, where l is the length of the edge, and c is set to 3.5 pixels for all the tested images. This model shows a better result than setting the noise magnitude as a constant value for all lines. Then, the vanishing hull is computed using the dual-space algorithm, and its centroid is used to find the approximated optimal estimation of the vanishing point. Figure 8 compares the result of the VH and ML method. As predicted by the graph of Figure 6 (a), the advantage of the VH method is apparent compared with the ML method for small angle images. Note that, even for images with such small angles, the VH method gives very good result without using the Gaussian sphere.

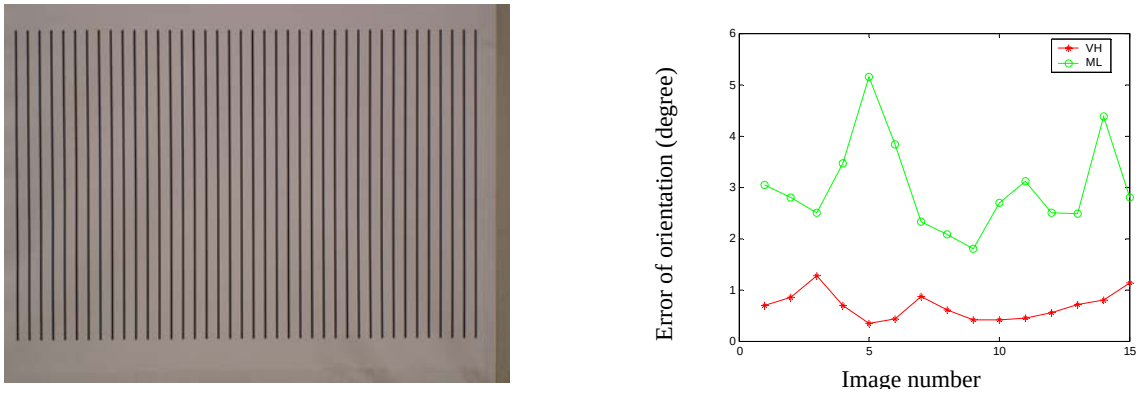
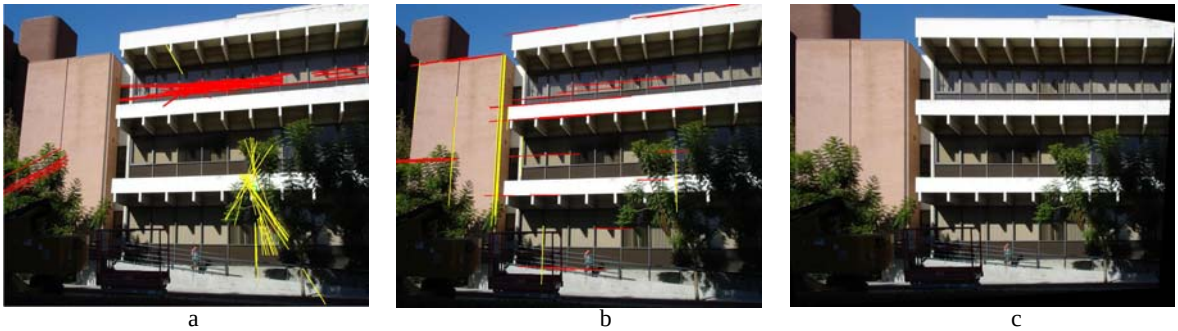


Figure 8. Real data comparison, indoor images. Left: one of the 15 images, right: performance comparison. The error of the VH method (red stars) is much smaller than the ML method (green circles), which shows the advantage of the VH method.

Outdoor images

For outdoor images, since it is hard to control the camera's motion in outdoor environments, we opt to use a manual verification by measuring the horizontal and vertical line angle error of the rectified images. We discuss two typical cases here, small and large rotation angle.

The first case is a strong perspective image with a large rotation angle (Figure 9 (a)). This image is also one of the typical test cases, where images have trees occlusions and small-scale textures. Our vanishing point detection method is robust to these difficulties. Thanks to the heuristic filters, spurious vanishing points caused by trees and small-scale textures (Figure 9 (a)) are filtered (Figure 9 (b)), which shows the effectiveness of the filter. The vanishing hull is found using the dual-space algorithm, then the vanishing point is computed as the centroid assuming a uniform distribution, finally the image is rectified (Figure 9 (c)). Careful manual user verification shows that the horizontal angle error of the VH method is less than 0.1 degrees (the standard deviation of the vanishing hull is $\sigma = 0.12$ degrees), and the vertical angle less



a b c

Figure 9. Our door images: a strong perspective image case. Our vanishing point detection method is robust to trees occlusions and small-scale textures. (a) Clustering result before filtering , (b) the spurious vanishing points are filtered using the heuristic filters, and the image is rectified (c)

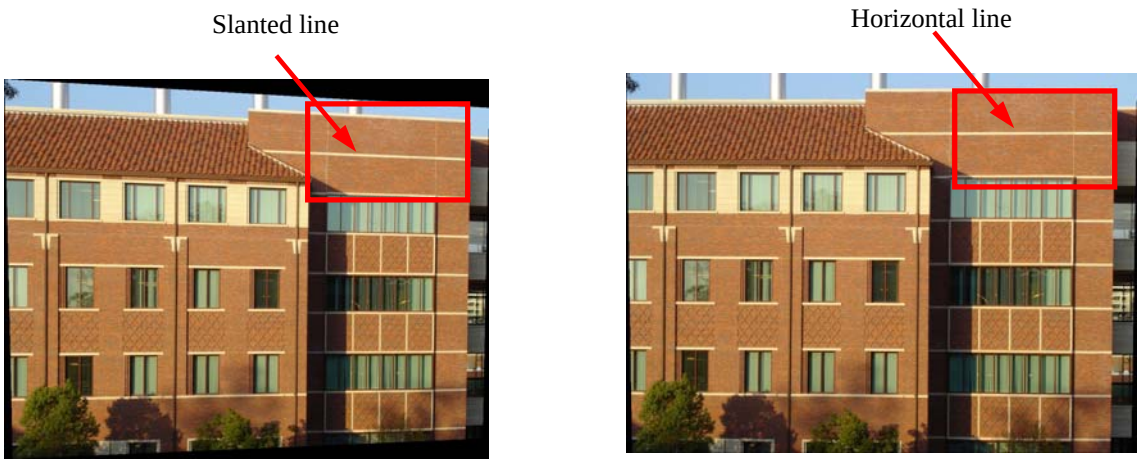


Figure 10. Compare rectified image of the VH and ML method for a weak perspective image. Left: ML method, the line that should be horizontal is slanted. Right: VH method, the line is correctly rectified.

than 0.15 degrees ($\sigma = 0.17$), while ML method gives 0.2 and 0.3 degrees angle error respectively. Both methods perform well, though the VH method is slightly better.

The second image is a weak perspective image with a small rotation angle (Figure 10). The result of the VH method shows that the *x* direction vanishing point is 2×10^6, and the y direction is -4×10^4. Careful manual user verification shows that the horizontal angle error of VH method is less than 0.3 degrees ($\sigma = 0.35$), and the vertical angle less than 0.2 degrees ($\sigma = 0.22$), while ML method gives 1.94 and 0.55 degrees angle error respectively[3]. Figure 10 compares the result of the rectified image using both methods, the VH method is apparently superior for this case, which is consistent with the graph of Figure 6 (a).

The comparison result is summarized in Table 2. Both typical cases show that the VH method gives better performance (optimal solution), and the user verified orientation error is within the range of variance (accuracy). Hence the true vanishing points are within the region of the vanishing hull. All the vanishing points are stable because the vanishing hulls are all closed convex polygons. Both simulation and real data show that our method is superior.

Table 2. Outdoor real images comparison.

Method	Strong Perspective Image (Orientation error, degree)		Weak Perspective Image (Orientation error, degree)	
	X	Y	X	Y
VH	0.1	0.15	0.3	0.2
ML	0.2	0.3	1.94	0.55

[3] For a line with length 570 pixels, 0.1 degree angle error corresponds to 1 pixel error.

7 Application

7.1 Generate façade textures

The vanishing hull concept is used to compute vanishing points, then recover poses for real images [38], and generate rectified textures. This technique has been used on hundreds of images to generate textures for a university campus, the results showing that the recovered poses are very accurate. Figure 11 shows a number of rectified textures and Figure 12 shows some rendered images of a university campus with dozens of textures mapped to 3D models. The rectified textures are visually very accurate, which shows the accuracy of the VH method.

7.2 Remove occlusions

We have also applied the VH method to recover poses for mages at different viewpoints, then remove occlusions and generate textures. As shown in Figure 13, both images ((a) and (b)) are occluded by a pole but at different places. The user interactively mark the pole using a rectangle (note that we do not require the user to mark the exact shape of the pole), then the algorithm automatically recovers the poses for each image using the VH method. Finally, we generate the rectified texture (Figure 13 (c)) by copying pixels from one image to another in the red rectangle area and blending in the other areas. Note that the pole is removed and the image is rectified.

7.3 Hybrid modeling

The vanishing hull concept is also used in hybrid modeling using both LiDAR and aerial images [39]. We first estimate the pose of an aerial image downloaded from the Internet (Figure 14 (a)) using the VH method, then project the image to a plane to generate an orthographic aerial image (Figure 14 (b)). An interactive method is developed to extract high-resolution outlines from the rectified aerial image, then we combine the height information from the LiDAR data [39] to generate models with accurate surfaces and edges (Figure 12). For further details, please refer to [39].

8 Conclusion

Vanishing points are valuable in many vision tasks such as orientation and pose estimation. This paper defines the concept of vanishing hull from a geometric viewpoint, which is the intersection of the edge regions. Based on the edge error model, we also present a novel geometric image grouping method for vanishing points detection. The vanishing hull gives the region of the true vanishing point, and its probability distribution determines the property of vanishing points. The expectation of the vanishing hull is the optimal solution of the vanishing point, its variance defines the accuracy of the estimation, and its shape determines the stability of the vanishing point. Extensive simulation and real data experiments show that our method is superior to one state-of-the-art technique. We also present many applications using the vanishing hull method.

The concept of the vanishing hull is derived from the edge error model, so its properties depends on the edge error model, which makes it a valuable tool to analyze the performance of different edge error models. So future work would be compare the performance of different edge error models using the vanishing hull concept.

Acknowledgment

This work made use of Integrated Media Systems Center Shared Facilities supported by the National Science Foundation under Cooperative Agreement No. EEC-9529152.

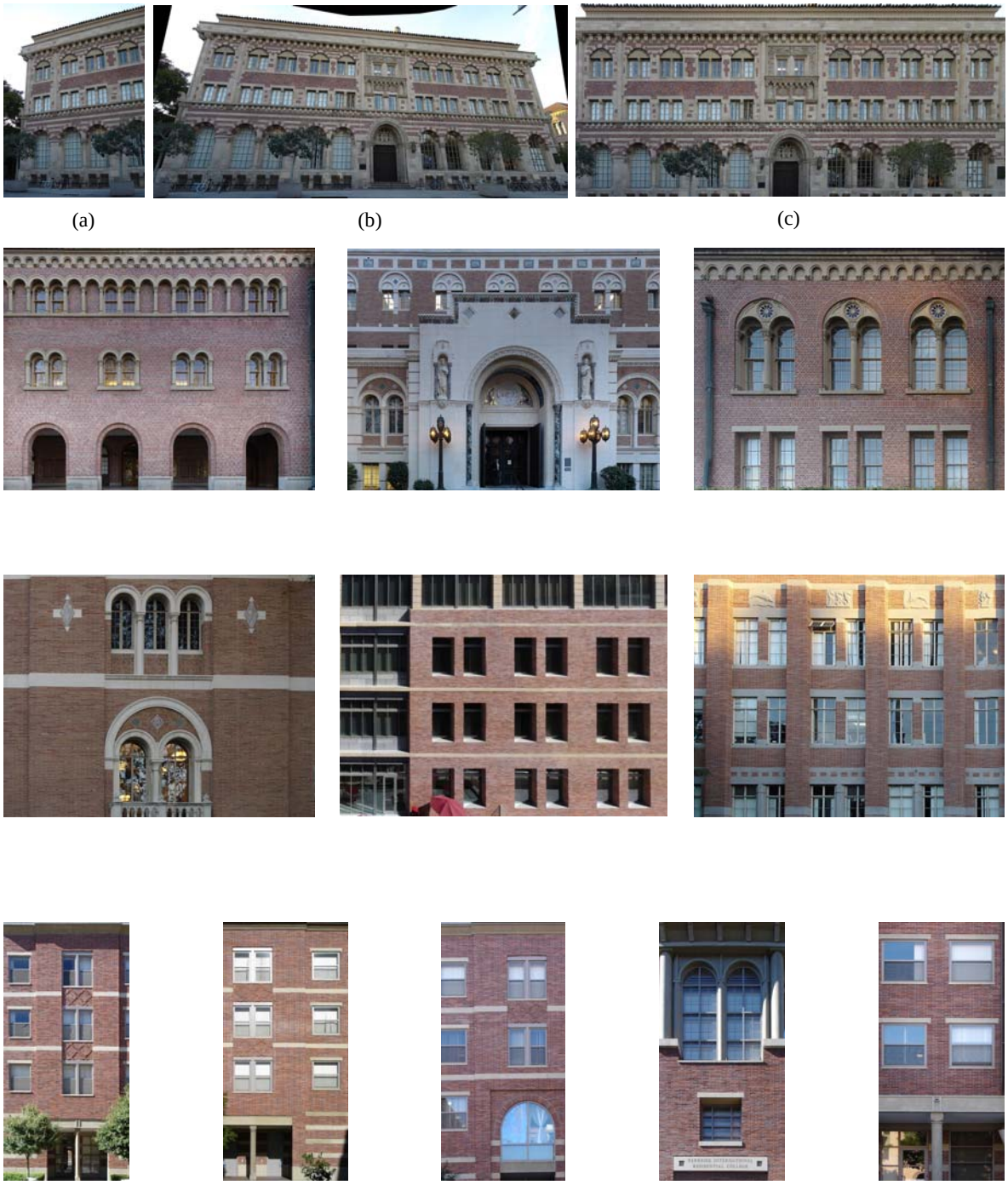


Figure 11. Generate textures. Vanishing hull is used to estimate vanishing points and compute poses, then rectify images to generate textures. One of the original images is shown (a), and the images are stitched as a mosaic (b). The pose is automatically estimated, and the mosaic is rectified to generate a high quality texture image (c). A number of textures rectified using our method are also shown. The rectified textures are visually very accurate.

Figure 12. Rendered images of dozens of textures mapped to 3D models of a university campus.

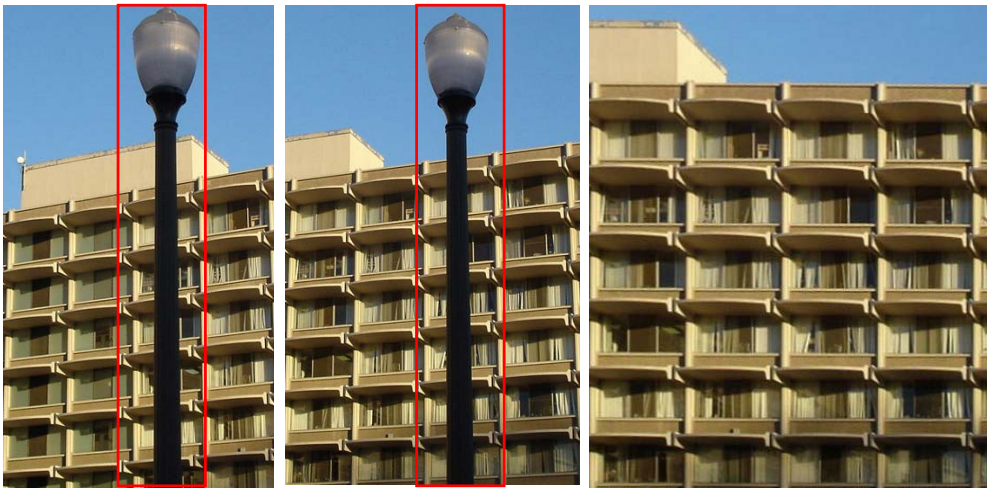


Figure 13. Remove occlusions. Both images ((a) and (b)) are occluded by a pole but at different places. The VH method is used to estimate poses for each image and generate rectified textures with occlusions removed (c).

Figure 14. Rectify an aerial image for hybrid modeling.

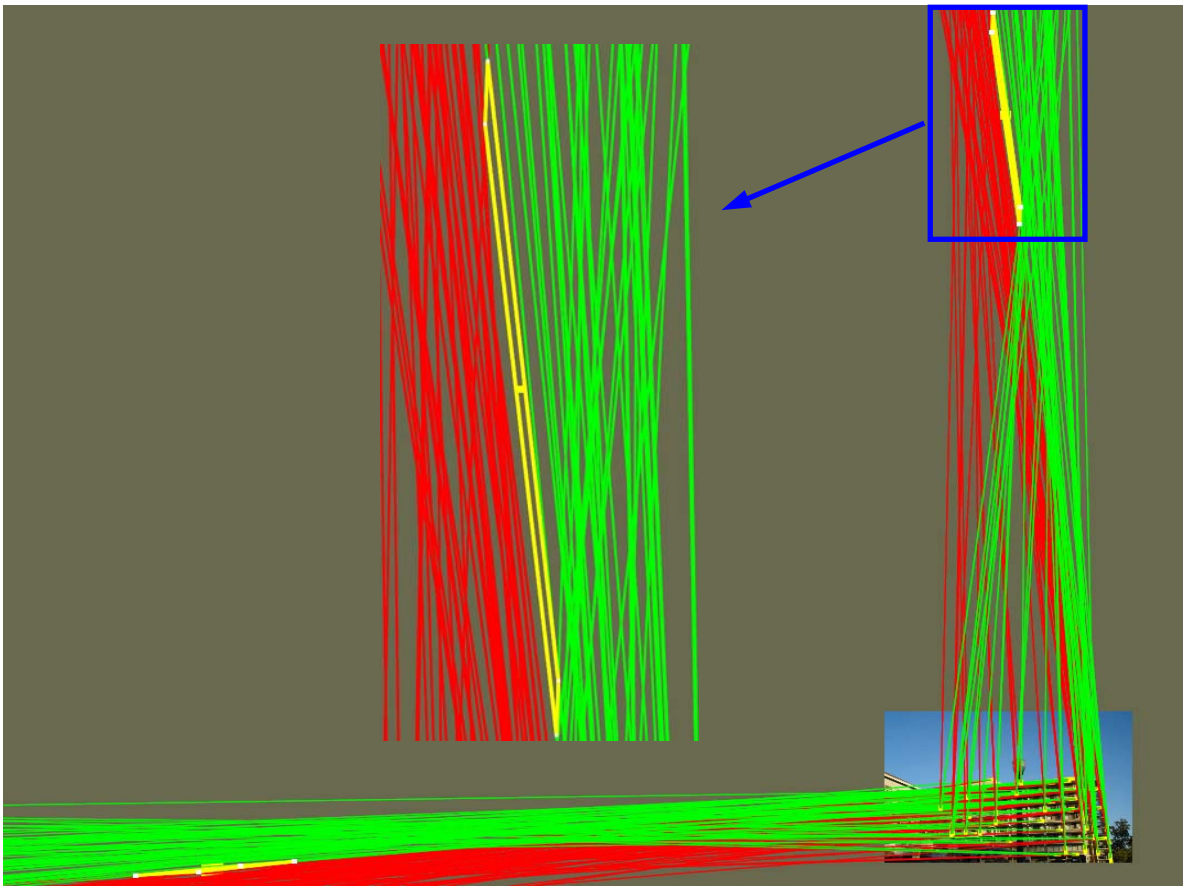


Figure 15. Vanishing hull of a real image. Each line (short yellow lines) in the image forms a fan shape edge region bounded by a red and a green line. These fan regions intersect at a convex polygon (yellow polygon), which is the vanishing hull. An enlarged image of the *y* direction vanishing hull is also shown, where the yellow dot in the center is the centroid of the vanishing hull.

References

[1] J. Shufelt, Performance evaluation and analysis of vanishing point detection techniques, IEEE Trans. Pattern Analysis and Machine Intelligence, 21 (3) (1999) 282–288.

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

[3] R. Cipolla, T. Drummond and D.P. Robertson, Camera calibration from vanishing points in images of architectural scenes, in: Proceedings of British Machine Vision Conference, 1999, pp.382–391.

[4] M. Antone and S. Teller, Automaic recovery of relative camera rotations for urban scenes, International Conference on Computer Vision and Pattern Recognition, 2000, pp.282–289.

[5] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master, Calibrated, registered images of an extended urban area, International Journal of Computer Vision, 53 (1) (2003) 93–107.

[6] R. Schuster, N. Ansari and A. Bani-Hashemi, Steering a robot with vanishing points, IEEE Transactions on Robotics and Automation, 9 (4) (1993) 491-498.

[7] A. Criminisi, I.D.Reid, and A. Zisserman, Single view metrology, International Journal of Computer Vision, 40 (2) (2000) 123-148.

[8] E. Guillou, D. Meneveaux, E. Maisel. and K. Bouatouch, Using vanishing points for camera calibration and coarse 3D reconstruction from a single image, The Visual Computer, 16 (2000) 396-410.

[9] D. Liebowitz, A. Criminisi and A. Zisserman, Creating architectural models from images, Computer Graphics Forum, 18 (3) (1999) 39–50.

[10] J. Hu, S. You and U. Neumann, Vanishing hull, Third International Symposium on 3D Data Processing, Visualization and Transmission, University of North Carolina, Chapel Hill, 2006.

[11] F.A. van den Heuvel, Vanishing point detection for architectural photogrammetry, International archives of photogrammetry and remote sensing, 32 (5) (1998) 652–659.

[12] R. T. Collins and R. S. Weiss, Vanishing point calculation as statistical inference on the unit sphere, International Conference on Computer Vision, 1990, pp.400-403.

[13] A. Almansa, A. Desolneux and S. Vamech, Vanishing point detection without any a priori information, IEEE Trans. Pattern Analysis and Machine Intelligence, 25 (4) (2003) 520–507.

[14] B. Brillault-O'Mahoney, New method for vanishing point detection, Computer Vision, Graphics, and Image Processing, 54 (1991) 289-300.

[15] M. Bosse, R. Rikoski, J. Leonard, and S. Teller, Vanishing points and 3D lines from omnidirectional video, The Visual Computer, Special Issue on Computational Video, 19 (6) (2003) 417-430.

[16] C. Burchardt and K. Voss, Robust vanishing point determination in noisy images, International Conference of Pattern Recognition, 2000.

[17] L. Grammatikopoulos, G.Karra and E. Petsa, Camera calibration combining images with two vanishing points, International Society for Photogrammetry and Remote Sensing, 2004.

[18] M.J. Magee and J.K. Aggarwal, Determining vanishing points from perspective images, Computer Vision, Graphics, and Image Processing, 26 (1984) 256–267.

[19] S. Heuel, W. Forstner, Matching, reconstructing and grouping 3D lines from multiple views using uncertain projective geometry, International Conference on Computer Vision and Pattern Recognition, 2001, Vol. II, pp.517-524.

[20] S. Heuel, Points, lines, and planes and their optimal estimation, Lecture Notes In Computer Science, Vol. 2191, Proceedings of the 23rd DAGM-Symposium on Pattern Recognition, 2001, pp.92 – 99.

[21] J.C. Clarke. Modelling uncertainty: A primer, Technical Report, 2161, University of Oxford, Dept. Engineering Science,1998.

[22] K. Kanatani, Statistical Analysis of focal-length calibration using vanishing points, IEEE Trans. Robotics and Automation, 8 (6) (1992) 767–775.

[23] P.H.S. Torr, Bayesian model estimation and selection for epipolar geometry and generic manifold fitting, International Journal of Computer Vision, 50 (1) (2002) 35-61.

[24] S. M. Seitz and P. Anandan, Implicit representation and scene reconstruction from probability density functions, International Conference on Computer Vision and Pattern Recognition, 1999, pp. 28–34.

[25] M. Barnabei, A.Brini and G-C. Rota, On the exterior calculus of invariant theory, Journal of Algebra, 96 (1985) 120-160.

[26] S. Carlsson, The double algebra: an effective tool for computing invariants in computer vision, in J. Mundy, A. Zisserman and D. Forsyth, editors, Applications of Invariance in Computer Vision, number 825 in LNCS. Springer, 1994.

[27] O. Faugeras and T. Papadopoulo, Grassmann-cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors, In Trans. of the Royal Society A, 365 (1998) 1123–1152.

[28] O. Faugeras and B. Mourrain, On the geometry and algebra of the point and line correspondences between N images, Technical Report 2665, INRIA, 1995.

[29] D. Liebowitz and A. Zisserman, Metric rectification for perspective images of planes, International Conference on Computer Vision and Pattern Recognition, 1998, pp.482–488.

[30] G. Mclean and D. Kotturi, Vanishing point detection by line clustering, IEEE Trans. Pattern Analysis and Machine Intelligence, 17 (11) (1995) 1090-1095.

[31] C. Rother, A new approach for vanishing point detection in architectural environments, in: Proceedings of British Machine Vision Conference, 2000.

[32] S. Barnard, Interpreting perspective images, Artificial Intelligence, 21 (1983) 435-462.

[33] E. Lutton, H. Maitre, and J. Lopez-Krahe, Contribution to the determination of vanishing points using hough transform, IEEE Trans. Pattern Analysis and Machine Intelligence, 16 (4) (1994) 430–438.

[34] G. Schindler and F. Dellaert, Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments, International Conference on Computer Vision and Pattern Recognition, 2004.

[35] J.M. Coughlan and A.L. Yuille. Manhattan world. Neural Computation, 2003.

[36] M. Berg etc. Computational geometry algorithms and applications. ISBN: 3-540-65620-0, published by Springer.

[37] E. Weisstein. Green's theorem, MathWorld, http://mathworld.wolfram.com/ GreensTheorem.htm.

[38] J. Hu, S. You and U. Neumann, Automatic pose recovery for high-quality textures generation. International Conference of Pattern Recognition, HongKong, 2006.

[39] J. Hu, S. You and U. Neumann, Integrating LiDAR, aerial image and ground images for complete urban building modeling, Third International Symposium on 3D Data Processing, Visualization and Transmission, University of North Carolina, Chapel Hill, 2006.

Vanishing Hull: A Geometric Concept for Vanishing Points Detection and Analysis

Jinhui Hu*, Suya You, Ulrich Neumann

University of Southern California, 3737Watt Way, PHE 404
Los Angeles, California 90089
Tel: 213-740-4250, Fax: 213-740-8931
{jinhuihu, suyay, uneumann}@graphics.usc.edu

Abstract

Vanishing points are valuable in many vision tasks such as orientation estimation, pose recovery and 3D reconstruction from a single image. This paper proposes a new concept from a geometric viewpoint, *vanishing hull*, which can be used to detect and quantitatively analyze the stability and accuracy of vanishing points. Given an edge error model, the range of a true edge can be modeled using a fan region. The geometric intersection of all these fan regions is a convex hull, which is called the vanishing hull. The vanishing hull gives the region of a true vanishing point, and its distribution determines the probability of the vanishing point. The expectation of the vanishing hull is the optimal solution of the vanishing point, its variance defines the accuracy of the estimation, and its shape determines the stability of the vanishing point. Hence, we can quantitatively analyze the stability and accuracy of the vanishing point estimation using the vanishing hull concept. Extensive simulation and real data results show that our method is significantly better than one state-of-the-art technique.

Keywords

Vanishing Point, Vanishing Hull, Edge Error Model

* Corresponding author. Email: jinhuihu@graphics.usc.edu. Tel: 1-213-740-4250, Fax: 1-213-740-8931.

1 Introduction

A vanishing point is defined as the intersection point of a group of image lines that correspond to the projection of parallel lines in 3D with an ideal pin-hole camera model. The position of a vanishing point in the image plane is only determined by the camera center and the orientation of the 3D lines in the camera system. Vanishing points are valuable in many vision tasks. A traditional application of vanishing points is building detection in aerial images [1]. Vanishing points can be used to group images lines, which are then used to form hypotheses of building edges. Cameras can be calibrated using three orthogonal vanishing points [2,3], and pose can be recovered [4,5]. Other applications of vanishing points include robot navigation [6] and 3D reconstructing from a single image [7,8,9]. The goal of this paper is to present a new consistent framework for vanishing points detection, and stability and accuracy analysis. We present extensions and new capabilities, and discuss theory and algorithm details that were omitted or summarized in our previous work [10].

1.1 Motivation

Much research has been conducted on accurately identifying the position of vanishing points. This varies from simple line grouping [11] to more complicated methods using statistical models [12]. Most previous research work focuses on finding the group of lines corresponding to valid vanishing points, or the vanishing points detection problem, and the performances are often evaluated empirically [1]. However, there is very little research focused on finding a theory to quantitatively analyze the stability and accuracy of vanishing points estimation. This is the main motivation of this work.

Different from most previous work [13-18], we attack this problem from a geometric viewpoint. Observing that the region spanned by lines sampled from an edge error model is a fan region (Figure 1, Section 2.1), we intersect all the fan regions to form a convex polygon, called the *Vanishing Hull*. This papers shows that the vanishing hull has some interesting properties and it can lead us to an optimal estimation of vanishing points. In more details, the vanishing hull gives the region of the true vanishing point, and its distribution gives the probability of the vanishing point. The expectation of the vanishing hull, the centroid for a uniform distribution, gives the optimal solution of the vanishing point under statistical meaning, its variance defines the accuracy of the vanishing point and its shape determines the stability of

the vanishing point. Hence, the vanishing hull concept provides a theoretical framework to quantitatively analyze the region, optimal solution, stability and accuracy of vanishing points. Besides a framework for analyzing vanishing points with image noises, we also present a novel geometric edge grouping method based on edge error models.

Before continuing, we would like to point out that we are not the first one to explore the idea of estimating the probability distribution of geometric entities in compute vision. Similar work has been done by Wolfgan Forstner and Stephan Heuel[19,20]. Forstner and Heuel combine geometry and statistics uncertainty [21-23] for line grouping [20] and 3D reconstruction [19,24]. While their work convert the geometric problem of joint and union into an algebra problem using double algebra [25-28], we are more focused on the geometric entities, such as the shape and centroid of the vanishing hull. More significantly, we avoid the usage of covariance propagation, which becomes cumbersome when the number of lines are large, a typical situation for vanishing points estimation. Forstner and Heuel's method is suitable for 3D reconstruction with pairs of lines, while our method is more suitable for vanishing points estimation with large number of lines[1].

1.2 Related work

There are two key problems in identifying vanishing points, finding the group of image lines that correspond to a true vanishing point and computing the position of the vanishing point with the presence of image noises. According to the methodology used in the two steps, we classify different methods into two classes, clustering methods and voting methods.

Clustering methods first find possible clusters using the intersection of all pairs of lines [29] or image gradient orientations [30]. Then a criterion of distance or angle is used to assign each line to different clusters. The drawbacks of clustering methods are the high computational complexity and that a hard threshold is needed to group lines into clusters. Liebowitz and Zisserman [29] first group the image lines, then use histogram to find the dominant vanishing points. A Maximum-Likelyhood estimator is used to compute the position of the vanishing point. McLena and Kotturi [30] integrate edge detection and line clustering to the process of vanishing points detection, and then use a non-linear method to compute the

[1] Our experiments show that the larger the number of lines, the more accurate the vanishing point estimation.

position of vanishing points with a statistical edge error model.

Voting methods can be classified into image space methods and Gaussian sphere methods according to the space they use for voting. Rother[31] accumulates the votes of image lines in the image space, then search the vanishing points using knowledge such as orthogonal criteria and camera criteria. However, this method is computational expensive.

A popular method is using the Hough Transform in Gaussian sphere space [32], which is a global feature extraction method [2]. Many improvements have been made to address the shortcomings of Hough-based approaches[12,33]. Shufelt [1] uses the knowledge of primitive models to reduce spurious maxima, and an edge error model to improve the robustness to image noises. The drawback of voting in Gaussian sphere is that the accuracy is limited to the discretization of the accumulator space, hence it is hard to achieve the precision that the image can provide. Antone and Teller [4,5,15] use a hybrid method of Hough Transform and least square to detect and compute vanishing points. G. Schindler and F. Dellaert [34] use a EM method to compute vanishing points in *Manhattan world* [35].

Recently, Almansa et al. [13] propose a system using vanishing regions of equal probability to detect vanishing points without any a priori information. A vanishing region is different from a vanishing hull, the former is an accumulation space for vanishing points detection while the latter is a tool to analyze the stability and accuracy of vanishing points.

The rest of the paper is organized as following. We first introduce the vanishing hull concept and its properties based on a simple end-points edge error model (Section 2), then we present some novel methods for vanishing points detection based on the edge error model (Section 3). Section 4 presents an algorithm to determine the vanishing hull and analyze vanishing points estimation, and we extend the vanishing hull concept to general edge error models in Section 5. The performance of our method is extensively analyzed with both simulation and real data, and quantitatively compared with one state-of-the-art technique [29](Section 6). Finally, we present some applications in Section 7 and conclude the paper in Section 8.

2 Vanishing Hull

Since the idea of vanishing hull is derived from the intersection of edge regions, we first present a simple edge error model, then introduce the definition and properties of the vanishing hull.

2.1 Edge error model

Various edge error models have been presented. McLean and Kotturi [30] use a statistical model to present both the error of the line centroid and the error of its orientation. Other models using both geometry and statistics can be found in [19]. Shufelt [1] presents a simple but effective edge error model. Inspired by the idea of a fan edge region, we derive the concept of vanishing hull by the intersection of all these regions. We first adapt this simple edge error model to our vanishing hull framework, and then extend the concept to general edge error models (Section 5).

Consider the representation of a line segment using two endpoints, and assume that the two end points have one pixel precision, then two fan regions with a rectangular region in the middle can be formed by moving the two end points freely in the pixel squares (Figure 1). This region is a concave region, so we cannot guarantee that the intersection of such regions will be convex. Fortunately, a true vanishing point cannot lie in the image line segment (Section 3.1), so the rectangular region has no effect on the shape of the intersection of edge regions. We simply take the middle point of the edge, and form two fan regions with the two end points. Furthermore, a true vanishing point can only lie in one direction of the edge, so we just take one of the fan region, which is a convex region (Figure 1).

2.2 Vanishing hull definition and property

The *Vanishing Hull* is defined as the intersection of the fan-shape edge regions with a given edge error model (Figure 2). Figure 15 shows a vanishing hull of a real image. We first present the properties of the vanishing hull, the computation and algorithm details are presented in section 4.

Property I. A vanishing hull is not empty. Proof: according to the grouping method that will be presented shortly in Section 3.1, an edge is assigned to a cluster only if its edge region covers the intersection point of the cluster, so the vanishing hull contains at least the intersection point.

Property II. A vanishing hull is convex. Proof: the intersection of convex regions (edge regions are fan shape regions, hence convex) is convex.

Property III. A true vanishing point lies inside the region of its vanishing hull with the assumption that the edge error model is correct (the true edge lies inside the fan region). Proof: By definition, the true vanishing point must lies inside the union of all the edge regions. Now assume the vanishing point *VP* lies outside of the vanishing hull but inside the union of the edge region. Then there must exist some edge, say

L, whose edge region does not cover *VP*. Hence the edge error model of edge *L* is wrong, which is contradictory with our assumption. Hence, *VP* must be inside the vanishing hull. This property is important, it tells us where to find the true vanishing point.

Property IV. The centroid of a vanishing hull is the optimal estimation of the vanishing point with a uniform distribution model. Proof: the optimal estimation of the vanishing point is the expectation of the probability distribution of the vanishing points inside the vanishing hull under statistical meaning. With a uniform distribution, the expectation of a vanishing hull is its centroid.

Property V. The variance of a vanishing hull determines the accuracy of the estimated vanishing point. Proof: this follows directly from the probability theory.

Property VI. The shape of a vanishing hull determines the stability of the estimation of the vanishing point.

A vanish hull can be open, it can also be a closed non-trivial convex polygon, a line segment or a point (Figure 3). When the image lines are parallel, the vanishing hull is an open convex hull (Figure 3 (a)), the centroid is undetermined, which means the estimation of the vanishing point is unstable. This is reasonable because edges have noises, any non-zero noise will be enlarged to infinity when the vanishing point is at infinity, which makes the estimation unreliable. An open vanishing hull indicates a vanishing point at infinity, which corresponds to a point at the great circle parallel to the image plane in the Gaussian sphere. We handle the open vanishing hull case by setting the vanishing point to infinity.

When the vanishing hull is a closed non-trivial convex polygon (Figure 3 (b)), the vanishing point can be estimated using the centroid with the variance of the distribution as the estimation accuracy. When the vanishing hull shrinks to a line segment (Figure 3 (c)), the uncertainty is along just one direction, and the vanishing point can be precisely computed when the vanishing hull degenerates to a point (Figure 3 (d)), which corresponds to an error-free edge model.

3 Vanishing Points Detection

Since we are interested in finding the intersection regions of all the edges, it is necessary to identify the image lines that can form a possible vanishing point, i.e., we need to group lines into different clusters to detect vanishing points. We describe the grouping process in both image space and the Gaussian sphere.

3.1 Image space

Finding clusters

The Canny edge detector is used to find sub-pixel edges, then Hough Transform is used to find possible lines, and nearby lines are connected using some user defined threshold. The lines are represented using two end points.

The intersections of all pairs of line segments are computed to find all possible clusters. The computational complexity is $O(n^2)$, where n is the number of lines. Grouping lines into different clusters takes O(n) time, so the overall time complexity is $O(n^3)$, which is expensive for a large number of lines. We will reduce the complexity using a filtering step and the RANSAC algorithm later.

Grouping

After finding the clusters, we need a criterion to assign lines to different clusters. A distance criterion gives priority to close vanishing points, while an angle criterion gives priority to far vanishing points. A reasonable threshold is to use a tuple of both distance and angle, or use normalized angle error [30]. However, all these methods need a hard threshold, which may be inconsistent with the edge error model.

We use a geometric grouping method that is consistent with the edge error model without any hard thresholds. For each cluster of two lines, we can find the intersection region A of the edges, and a test edge is assigned to this cluster when its edge region overlaps with region A. Furthermore, we use a strong constraint for clustering (Figure 4). An edge is assigned to a cluster only if its edge region covers the intersection point of the cluster. This guarantees that the intersection region of the edge regions in each cluster is not empty (Property I in Section 2.2). The normalized length of each edge is accumulated in its assigned cluster, and the maximum clusters are chosen to compute potential vanishing points.

Filtering spurious vanishing points

Most of our testing images are outdoor building images with heavy occlusion by trees (Figure 9 (a)), which causes many spurious vanishing points. Knowledge of the image and vanishing points are used to filter spurious vanishing points. First we roughly classify the extracted lines into x and y groups[2] according

[2] Each group may contain more than one vanishing point.

to the line orientation to reduce the size of line number. Then vanishing points are filtered using the following three filters.

1) Iterative line length. According to the edge error model, longer lines are more reliable, however, we would like also to keep shorter lines. So we first filter the lines using a large length threshold, then estimate the possible vanishing points, and these points are used to find more short line supporters according to the grouping method.

2) Covering area. Another observation of the image is that edges of trees only cover a small part of the image region, so the ratio of the covering area against the image area is also used to filter spurious vanishing points.

3) Valid vanishing point. Vanishing points are the intersection of image lines that correspond to parallel lines in 3D. So by definition, a valid vanishing point will not lie on the image segment in the image space. This filter is very effective in reducing spurious clusters.

RANSAC

Even though we classify lines into two groups to reduce the line number, and use filters to reject spurious clusters, the number of clusters may still be large. Since we are interested in find vanishing points that correspond to dominant directions, the RANSAC algorithm is used to find the maximum cluster of lines. Generally there exist three orthogonal vanishing points for images of outdoor buildings. We first find the dominant vanishing points for x and y direction. The vanishing point of z direction is estimated using the orthogonal property of the three directions, and its supporting lines are found using our grouping method to refine the position using the vanishing hull.

3.2 Gaussian sphere

The geometric grouping method can also be adapted to the Gaussian sphere space. According to the edge error model, each line casts a swath on the Gaussian sphere rather than a great circle. Similar to Shufelt [1], we can rasterize the swath into the Gaussian sphere using polygon boundary fill algorithm. The three heuristic filters can be adapted as well. For the iterative line length filter, we first only scan lines with length above some threshold into the Gaussian sphere, then discretize the sphere and collect votes weighted with line length. Then we find the dominant cell and scan more short lines to find more supporting lines.

For the other two filters, we find the corresponding lines for each dominant cell, and compute the covering area and filter them with a threshold, and finally test whether the dominant cells lie out of each line segment.

The Gaussian sphere method has the advantage of treating each vanishing point equally, including infinite vanishing point. However, in experiments with real images, we opt to use the clustering method in image space rather than the Gaussian sphere for several reasons. First, the accuracy of the Gaussian sphere is limited to the discretization accuracy, hence it is hard to achieve the precision that an image can offer. Secondly, the intersection of the fan regions of the edges that belong to the maximum cell in the Gaussian sphere may be empty, which makes the vanishing hull meaningless. While in the image space, we can guarantee the vanishing hull to be non-empty using the aforementioned grouping method. The last reason is that the projected vanishing hull onto the Gaussian sphere is not a polygon anymore, so it is hard to compute the probability distribution and analyze the stability and accuracy. Furthermore, we argue that we treat finite and infinite vanishing points equally even using image space grouping method. This is due to Property VI of the vanishing hull. When the vanishing point is finite, it can be well determined by the vanishing hull. When the vanishing point is at infinity, the vanishing hull is open, which can be easily detected. We set the orientation angle for an infinite vanishing point to zero.

4. Determining Vanishing Hull

4.1 Half-plane intersection

Given the group of lines that form a vanishing point, let us consider how to find the vanishing hull. A fan shape edge region can be considered as the intersection of two half planes (Figure 1), so the problem of finding the intersection of the edge regions can be cast as the problem of half-planes intersection. A naïve way to solve the problem is to add one half-plane bound line at a time to compute the intersection region, which takes $O(n^2)$ time. However, a more elegant algorithm with $O(n \lg n)$ time can be presented by utilizing the properties of dual space [36].

There exists an interesting property called duality between lines and points [36]. Given a non-vertical line L, it can be expressed using two parameters (k, b). We can define a point with k as the x coordinate and

b as the y coordinate. This point is called the dual point (L^*) of line L. The space of the lines is called the prime space, while the space of the points is called the dual space. The half-plane intersection problem in the prime space can be mapped as the problem of finding the convex hull of the points in the dual space.

We first divide the bound lines of half-planes into two sets: an upper set (half-planes lie above the bound lines) and a lower set (half-planes lie below the bound lines). Let's consider the upper set first. For each line L in the upper set, we can find a dual point $L^*(k, b)$. The intersection of all the half-planes in the upper set can be mapped as finding the lower convex hull (the edges of the lower boundary of the convex hull) of the dual points. The proof is out of the scope of this paper. Readers may refer to [36]. Similarly, we can find the intersection of all the half-planes in the lower set by determining the upper convex hull of the dual points. Note that according to our dual mapping, the upper convex hull and the lower convex hull will not intersect although the upper and lower half-planes do intersect, and that is the reason we split the half planes into two sets. Finally, the two regions are merged to find the vanishing hull. The algorithm is summarized as following:

1. *Split the half plane into two sets, an upper set and a lower set.*
2. *Map each set into a dual space and find the corresponding upper and lower convex hull (note that the two convex hulls are different).*
3. *Map the two half convex hulls back to prime space, and merge them to find the vanishing hull.*

Finding a convex hull of a point sets is a well-defined problem [36], which takes $O(n\lg n)$, with n as the number of points. The mapping and merging takes linear time, so the overall time complexity is $O(n\lg n)$.

4.2 Expectation and variance of vanishing hull

As aforementioned, we can estimate the position of the vanishing point using the expectation of the vanishing hull and analyze the accuracy using its variance. Now let us consider the computation of the expectation and variance of the vanishing hull. With a uniform probability distribution, the expectation and variance of the position of a true vanishing point can be computed using Equation 1, where D is the region of the vanishing hull, and A is its area. We only show x coordinate here, y coordinate can be computed in a similar way. Given a list of vertices of the vanishing hull, it is easy to show that the mean can be computed using the coordinates of these vertices (Equation 2, 3).

$$\mu(x) = \frac{1}{A}\iint_D x dA \qquad \mathrm{var}(x) = \frac{1}{A}\iint_D (x-\mu(x))^2 dA \tag{1}$$

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \tag{2}$$

$$\mu(x) = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{3}$$

The variance of the vanishing hull can also be represented as a simple expression of the coordinates of the vertices. According to Green's theorem [37], an integral in the region can be converted to an integral on the boundary:

$$\int_{\partial D} f(x,y)dx + g(x,y)dy = \iint_D (\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y})dxdy \tag{4}$$

Let $f = 1$, $g = \frac{1}{3}(x-\mu(x))^3$, we have

$$\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y} = (x-\mu(x))^2 \tag{5}$$

So: $\mathrm{var}(x) = \frac{1}{A}\iint_D (x-\mu(x))^2 dA = \frac{1}{A}\iint_D (\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y})dA$

$$= \frac{1}{A}\int_{\partial D} f(x,y)dx + g(x,y)dy \tag{6}$$

Let us consider the integral on line (x_i, y_i) to (x_{i+1}, y_{i+1}), we can parameterize the point on the line using t, such that:

$$x = x_i + t(x_{i+1} - x_{i)} \qquad y = y_i + t(y_{i+1} - y_{i)} \quad t \in [0,1] \tag{7}$$

Let $a_i = (x_i - \mu(x))$, $b_i = (x_{i+1} - x_i)$, $c_i = (y_{i+1} - y_i)$, it is easy to show that the variance of x coordinate can be computed as:

$$\mathrm{var}(x) = \frac{1}{3}\sum_{i=0}^{n-1} c_i (a_i^3 + \frac{3}{2} a_i^2 b_i + a_i b_i^2 + \frac{b_i^3}{4}) \tag{8}$$

Similarly, we can compute the variance of y coordinate.

5 Vanishing Hull for General Edge Error Model

The vanishing hull concept can be easily extended to general statistical edge error models. First we show an augmented vanishing hull considering the full edge region of the simple edge error model, then we show that a general vanishing hull can be derived in a similar way.

In section 2.1 we ignored the rectangular region, and used one of the edge fan to derive the vanishing hull concept. We claimed that the shape of the vanishing hull will not change with this approximation, which is true. However, the probability distribution in the vanishing hull is not a uniform distribution. For a full edge span region, an angle (called extreme angle θ) is formed by the vanishing point VP and two extreme points P_1 and P_2 (Figure 5). According to the edge error model, the two end points have equal probability inside the one-pixel-size square. The probability of a true edge passing through the vanishing point VP is determined by the overlapping area of the extreme angle and the pixel squares.

Let $p_i(l_i, VP)$ be the probability of a true edge l_i passing through VP, e_1 and e_2 be the two end points of l_i, and S_1 and S_2 be the overlapping region of the extreme angle with the two pixel squares, then:

$$p_i(l_i, VP) = p(e_1 \in S_1 \,\&\, e_2 \in S_2) \tag{9}$$

Assuming the two end points are independent with probability density function (PDF) $f(x,y)$ and $g(x',y')$ respectively, then the joint PDF is $f(x,y) * g(x',y')$, which is a 4D uniform distribution. Then $p_i(l_i, VP)$ is the integral of the joint PDF over the region $S_1\ S_2$.

$$p_i(l_i, VP) = \iint_{e_1 \in S_1 \,\&\, e_2 \in S_2} f(x,y) * g(x',y') dS_1 dS_2 \tag{10}$$

Note that points e_1, e_2 and VP are collinear. We can use a line-sweeping method to compute the integral in a region. Now consider a sweep line L_i that passes VP (Figure 5), and intersects the two squares with line segment l_1 and l_2 (Note that L_i is a line inside the extreme angle, while l_i is a true edge inside the whole edge region). Line L_i sweeps the whole overlapping region when its angle varies from 0 to θ, the integral of the joint PDF over the two line segments is:

$$p(e_1 \in l_1 \,\&\, e_2 \in l_2) = \int_{e_1 \in l_1 \,\&\, e_2 \in l_2} f(x,y) * g(x',y') dl_1 dl_2 \tag{11}$$

Denote the end points of line segment l_1 as $(x_1, y_1)(x_2, y_2)$ and l_2 as $(x_{1'}, y_{1'})(x_{2'}, y_{2'})$, they form a line segment in a 4D space (because we have three linear constraints: l_1, l_2 and they have the same slope). The integral of a uniform distribution over a line segment in 4D is just the length of the line segment, hence:

$$p(e_1 \in l_1 \,\&\, e_2 \in l_2) = [(x_2 - x_1)^2 + (y_2 - y_1)^2 + (x_{2'} - x_{1'})^2 + (y_{2'} - y_{1'})^2]^{\frac{1}{2}} \tag{12}$$

Now we can integrate over the angle to get the region integral:

$$p_i(l_i, VP) = \int_{\vartheta} p(e_1 \in l_1(\vartheta) \,\&\, e_2 \in l_2(\vartheta)) d\vartheta \tag{13}$$

The full expression of the analytical PDF of the vanishing hull for a full edge region is complicated, and the distribution function may not be continuous over the entire region. Even when the distribution for a single edge is continuous, the overall PDF is very high order due to the large number of edges. An analytical solution to the integral of such a high order PDF is very complicated, and may not exist. We use a discretizing method to solve this problem. The discretization process can achieve high precision (one pixel) because the vanishing hull region is bounded. Consider the center point, $VP(x, y)$, of a cell with one pixel size, we can find the extreme angle relative to edge l_i, and then compute the probability $p_i(l_i, VP(x, y))$ according to Equation 13. Then the probability of the point $VP(x, y)$ over all the edges is computed and normalized (Equation 14). The expectation and variance can be easily computed in the discretized vanishing hull.

$$P(x, y) = \prod_{i=0}^{n-1} p_i(l_i, VP(x, y)) \qquad P^*(x, y) = \frac{P(x, y)}{\sum_D P(x, y)} \tag{14}$$

Such a vanishing hull considering the full edge span region is called "augmented vanishing hull". In practice, we found that the vanishing hull often consists of only a few vertices (less than 10 vertices for 1000 lines), which means that the probability of a vanishing point being close to the edge region's boundary is very low. Since vanishing points close to the middle of the edge region has similar overlapping area with the pixel squares, it is reasonable to assume a uniform distribution for the vanishing hull formed of full edge regions.

We can extend the vanishing hull concept to general edge error model in a way similar to the augmented vanishing hull. A general edge error model often models the error of the edge centroid and orientation, the two end points or edge pixels, using a Gaussian distribution. The edge span region is still a fan shape, so the intersection of the edge regions is a convex hull. Assuming the PDF of the line l_i passing a vanishing point (x, y) is $f_i(x, y)$, then the PDF of the vanishing hull over all the lines is $\prod_i f_i(x, y)$. Again, this function is a high order non-linear function, we discretize the vanishing hull, and compute the mean and variance of the vanishing hull.

6 Analysis and Results

We have extensively analyzed the vanishing hull concept using both synthetic and real data. The performance of our method is also compared to one state-of-the-art algorithm [29].

6.1 Simulation data

We first analyze the vanishing hull concept using synthetic data. The goal of the simulation is to show that a vanishing hull gives the region of the true vanishing point, its expectation is the optimal solution, its shape determines the stability and its variance determines the accuracy of the estimation. The simulation is designed as following. A group of 3D parallel lines are projected by an ideal pin-hole camera to an image plane, then random noises with specified magnitude are added to the end points. The vanishing point is estimated using the centroid of the vanishing hull assuming a uniform distribution. We extensively analyze our Vanishing Hull (VH) algorithm with different parameter settings (Table 1). For each of the five groups of parameter settings, we sample the space with 100 evenly distributed intervals, the other four parameters are set as constant when one parameter varies in its range to test the performance relative to each parameter.

Vanishing hull is the true vanishing point region

The simulation shows that all the true vanishing points lie inside the vanishing hull. This is logical because the maximum noise magnitude is specified, so the edge error model exactly predicts the region of the true edge region, hence the true vanishing points lie inside the vanishing hull.

Expectation is the optimal solution

The result of VH method is compared with other two methods, Least Square (LS) method and Maximum Likelyhood (ML) method [29], to show that the expectation is the optimal solution. The comparison criterion is the recovered orientation angle error relative to the ground truth.

LS method uses least square to find the vanishing point closest to all lines, and ML method uses a non-linear method to minimize the distance of the line that passes the vanishing point and the mid-point to the two end points. According to our implementation, the difference of the ML and LS method is often several pixels, so the angle difference is very small. This is because ML method uses a non-linear optimization method, which often gives a local minimum close to the result of LS method. We just show the result of VH and ML method to make the figure clear (Figure 6).

The parameter θ and fov are related to the perspective effect of images. The result (Figure 6 (a), (b)) shows that the ML method gives large errors for weak perspective images (up to 40 degrees when the orientation angle is less than 0.1 degrees), while VH method performs reliably with maximum angle error less than 0.5 degrees and average less than 0.1 degrees. When the perspective effect is strong (orientation angle is larger than 10 degrees), the vanishing hull shrinks to several dozens of pixels, both methods perform well.

The parameter l and ε are related to the quality of edges. The simulation (Figure 6 (c), (d)) shows that the ML method gives large orientation errors for poor quality edges ($l < 30$ or $\varepsilon > 0.2$), and the maximum angle error is more than 14 degrees. The VH method performs significantly better than the ML method and very reliably over the whole range (maximum angle error 0.3 degrees, and average angle error less than 0.1 degrees).

The last parameter (Figure 6 (e)) compares the performance against the number of image lines. The simulation shows that the number of lines has no strong effects on the performance of ML method, however, it affects the performance of the VH method. When the number is more than 500, VH method performs very reliably, when the number drops to 100, it still performs significantly better than the ML method. However, when the number of lines drops below 50, the result is mixed. There are several reasons for this. First, the region of a vanishing hull shrinks with the increasing number of lines, so it is more reliable for more lines. Second, the expectation of the vanishing hull is the optimal solution for a vanishing

point under statistical meaning. However, when the sample (number of lines) is small, the true value of the vanishing point may deviate from the statistical value. The last reason is that we use a uniform distribution, which is an approximation as we showed in section 5.

In general, the VH method performs significantly better than the other two methods, especially for weak perspective images, and the performance of the VH method is very reliable with several hundreds of lines, a reasonable number for high-resolution images of buildings and aerial images. This shows that the expectation of the vanishing hull is the optimal solution of vanishing points estimation.

Stability and accuracy

We can also predict unstable vanishing points using the VH method. For most of the cases, the vanishing hull is closed, which indicates that the vanishing point is stable. When the vanishing hull is open, it indicates that the vanishing point is at infinity. The algorithm simply sets the orientation angle to zero for infinite vanishing points. We visualize unstable vanishing points with blue color in Figure 6 (f), where the ground truth $\theta = 0.01$. By setting the orientation of unstable vanishing points to zero, the VH method has a small error of 0.01 degree, while the ML method gives a large error (more than 10 degrees). Note that the VH method achieves an average error less than 0.5 degrees even for such ill-conditioned cases. The error of the VH method is also within the magnitude of the variance for all five groups of parameters, which implies that the variance determines the accuracy of the estimation. The graph of the variance is not shown here.

Since the noises in real images are generally Gaussian noises, we have also tested the performance of our algorithm with the Gaussian noises. Similar to the random noise case, we project a group of 3D parallel lines with an ideal pin-hole camera to an image plane, then add Gaussian noises with zero mean and specified variance to the end points. Again, we test the performance with five groups of parameter settings (Table 1). The vanishing point is estimated using the centroid of the vanishing hull rather than discretizing the vanishing hull and computing the exact distribution. It is interesting to note that, although the distribution is not uniform in the vanishing hull with the Gaussian noises, the centroid is still a very good approximation to the optimal solution. The reason might be that the optimal solution is close to the centroid with zero mean Gaussian noises. Of course, one can also compute the exact optimal solution using Equation 14, however, we find that the centroid approximation is good enough and much simpler to

compute. We also compared the performance of VH method with the ML method (Figure 7), the result is similar to that of the random noises.

6.2 Real data

The vanishing hull concept is also extensively tested with real images to generate textures (Section 7). However, the comparison of the performances of different algorithms in computing vanishing points for real images is difficult. The main challenge is that the ground truth positions of vanishing points for real images are unknown. Furthermore, the errors may come from different sources, such as the error of edge model, grouping errors, camera lens distortion and camera calibration errors. Here we compare the performance of the VH and ML method with two sets of images, indoor and outdoor images. More applications of the VH method for real images are presented in section 7.

Indoor images

For the indoor case, we print some pattern of parallel lines, and carefully place the camera with a tripod so that the image plane is parallel to the pattern plane. We translate the camera to take 15 different pictures while keeping the orientation unchanged. Figure 8 shows one of such images. By carefully controlling the camera motion, we can assume that the orientation ground truth is close to zero (careful user verification using the images shows that the variance is within 0.5 degrees). We have also tried to control the camera motion to take images with a slanted angle, however, it is very hard to measure the exact orientation angle. Furthermore, as shown in the graph of Figure 6 (a), when the rotation angle is more than 10 degrees, although the VH method still performs better than the ML method, the difference is so small that it will be covered by the measurement error.

To reduce the error from the camera, we calibrate the camera and correct the lens distortion for all the 15 images. As aforementioned, the vanishing hull concept is derived from the edge error model, so its performance depends on the edge error model. For real images, we model the noise magnitude of the end points of edges using an empirical model: $\varepsilon = c/\sqrt{l}$, where l is the length of the edge, and c is set to 3.5 pixels for all the tested images. This model shows a better result than setting the noise magnitude as a constant value for all lines. Then, the vanishing hull is computed using the dual-space algorithm, and its

centroid is used to find the approximated optimal estimation of the vanishing point. Figure 8 compares the result of the VH and ML method. As predicted by the graph of Figure 6 (a), the advantage of the VH method is apparent compared with the ML method for small angle images. Note that, even for images with such small angles, the VH method gives very good result without using the Gaussian sphere.

Outdoor images

For outdoor images, since it is hard to control the camera's motion in outdoor environments, we opt to use a manual verification by measuring the horizontal and vertical line angle error of the rectified images. We discuss two typical cases here, small and large rotation angle.

The first case is a strong perspective image with a large rotation angle (Figure 9 (a)). This image is also one of the typical test cases, where images have trees occlusions and small-scale textures. Our vanishing point detection method is robust to these difficulties. Thanks to the heuristic filters, spurious vanishing points caused by trees and small-scale textures (Figure 9 (a)) are filtered (Figure 9 (b)), which shows the effectiveness of the filter. The vanishing hull is found using the dual-space algorithm, then the vanishing point is computed as the centroid assuming a uniform distribution, finally the image is rectified (Figure 9 (c)). Careful manual user verification shows that the horizontal angle error of the VH method is less than 0.1 degrees (the standard deviation of the vanishing hull is $\sigma = 0.12$ degrees), and the vertical angle less than 0.15 degrees ($\sigma = 0.17$), while ML method gives 0.2 and 0.3 degrees angle error respectively. Both methods perform well, though the VH method is slightly better.

The second image is a weak perspective image with a small rotation angle (Figure 10). The result of the VH method shows that the x direction vanishing point is 2×10^6, and the y direction is -4×10^4. Careful manual user verification shows that the horizontal angle error of VH method is less than 0.3 degrees ($\sigma = 0.35$), and the vertical angle less than 0.2 degrees ($\sigma = 0.22$), while ML method gives 1.94 and 0.55 degrees angle error respectively[3]. Figure 10 compares the result of the rectified image using both methods, the VH method is apparently superior for this case, which is consistent with the graph of Figure 6 (a).

The comparison result is summarized in Table 2. Both typical cases show that the VH method gives better performance (optimal solution), and the user verified orientation error is within the range of variance

[3] For a line with length 570 pixels, 0.1 degree angle error corresponds to 1 pixel error.

(accuracy). Hence the true vanishing points are within the region of the vanishing hull. All the vanishing points are stable because the vanishing hulls are all closed convex polygons. Both simulation and real data show that our method is superior.

7 Application

7.1 Generate façade textures

The vanishing hull concept is used to compute vanishing points, then recover poses for real images [38], and generate rectified textures. This technique has been used on hundreds of images to generate textures for a university campus, the results showing that the recovered poses are very accurate. Figure 11 shows a number of rectified textures and Figure 12 shows some rendered images of a university campus with dozens of textures mapped to 3D models. The rectified textures are visually very accurate, which shows the accuracy of the VH method.

7.2 Remove occlusions

We have also applied the VH method to recover poses for mages at different viewpoints, then remove occlusions and generate textures. As shown in Figure 13, both images ((a) and (b)) are occluded by a pole but at different places. The user interactively mark the pole using a rectangle (note that we do not require the user to mark the exact shape of the pole), then the algorithm automatically recovers the poses for each image using the VH method. Finally, we generate the rectified texture (Figure 13 (c)) by copying pixels from one image to another in the red rectangle area and blending in the other areas. Note that the pole is removed and the image is rectified.

7.3 Hybrid modeling

The vanishing hull concept is also used in hybrid modeling using both LiDAR and aerial images [39]. We first estimate the pose of an aerial image downloaded from the Internet (Figure 14 (a)) using the VH method, then project the image to a plane to generate an orthographic aerial image (Figure 14 (b)). An interactive method is developed to extract high-resolution outlines from the rectified aerial image, then we

combine the height information from the LiDAR data [39] to generate models with accurate surfaces and edges (Figure 12). For further details, please refer to [39].

8 Conclusion

Vanishing points are valuable in many vision tasks such as orientation and pose estimation. This paper defines the concept of vanishing hull from a geometric viewpoint, which is the intersection of the edge regions. Based on the edge error model, we also present a novel geometric image grouping method for vanishing points detection. The vanishing hull gives the region of the true vanishing point, and its probability distribution determines the property of vanishing points. The expectation of the vanishing hull is the optimal solution of the vanishing point, its variance defines the accuracy of the estimation, and its shape determines the stability of the vanishing point. Extensive simulation and real data experiments show that our method is superior to one state-of-the-art technique. We also present many applications using the vanishing hull method.

The concept of the vanishing hull is derived from the edge error model, so its properties depends on the edge error model, which makes it a valuable tool to analyze the performance of different edge error models. So future work would be compare the performance of different edge error models using the vanishing hull concept.

Acknowledgment

This work made use of Integrated Media Systems Center Shared Facilities supported by the National Science Foundation under Cooperative Agreement No. EEC-9529152.

References

[1] J. Shufelt, Performance evaluation and analysis of vanishing point detection techniques, IEEE Trans. Pattern Analysis and Machine Intelligence, 21 (3) (1999) 282–288.

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

[3] R. Cipolla, T. Drummond and D.P. Robertson, Camera calibration from vanishing points in images of architectural scenes, in: Proceedings of British Machine Vision Conference, 1999, pp.382–391.

[4] M. Antone and S. Teller, Automaic recovery of relative camera rotations for urban scenes, International Conference on Computer Vision and Pattern Recognition, 2000, pp.282–289.

[5] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master, Calibrated, registered images of an extended urban area, International Journal of Computer Vision, 53 (1) (2003) 93–107.

[6] R. Schuster, N. Ansari and A. Bani-Hashemi, Steering a robot with vanishing points, IEEE Transactions on Robotics and Automation, 9 (4) (1993) 491-498.

[7] A. Criminisi, I.D.Reid, and A. Zisserman, Single view metrology, International Journal of Computer Vision, 40 (2) (2000) 123-148.

[8] E. Guillou, D. Meneveaux, E. Maisel. and K. Bouatouch, Using vanishing points for camera calibration and coarse 3D reconstruction from a single image, The Visual Computer, 16 (2000) 396-410.

[9] D. Liebowitz, A. Criminisi and A. Zisserman, Creating architectural models from images, Computer Graphics Forum, 18 (3) (1999) 39–50.

[10] J. Hu, S. You and U. Neumann, Vanishing hull, Third International Symposium on 3D Data Processing, Visualization and Transmission, University of North Carolina, Chapel Hill, 2006.

[11] F.A. van den Heuvel, Vanishing point detection for architectural photogrammetry, International archives of photogrammetry and remote sensing, 32 (5) (1998) 652–659.

[12] R. T. Collins and R. S. Weiss, Vanishing point calculation as statistical inference on the unit sphere, International Conference on Computer Vision, 1990, pp.400-403.

[13] A. Almansa, A. Desolneux and S. Vamech, Vanishing point detection without any a priori information, IEEE Trans. Pattern Analysis and Machine Intelligence, 25 (4) (2003) 520–507.

[14] B. Brillault-O'Mahoney, New method for vanishing point detection, Computer Vision, Graphics, and Image Processing, 54 (1991) 289-300.

[15] M. Bosse, R. Rikoski, J. Leonard, and S. Teller, Vanishing points and 3D lines from omnidirectional video, The Visual Computer, Special Issue on Computational Video, 19 (6) (2003) 417-430.

[16] C. Burchardt and K. Voss, Robust vanishing point determination in noisy images, International Conference of Pattern Recognition, 2000.

[17] L. Grammatikopoulos, G.Karra and E. Petsa, Camera calibration combining images with two vanishing points, International Society for Photogrammetry and Remote Sensing, 2004.

[18] M.J. Magee and J.K. Aggarwal, Determining vanishing points from perspective images, Computer Vision, Graphics, and Image Processing, 26 (1984) 256–267.

[19] S. Heuel, W. Forstner, Matching, reconstructing and grouping 3D lines from multiple views using uncertain projective geometry, International Conference on Computer Vision and Pattern Recognition, 2001, Vol. II, pp.517-524.

[20] S. Heuel, Points, lines, and planes and their optimal estimation, Lecture Notes In Computer Science, Vol. 2191, Proceedings of the 23rd DAGM-Symposium on Pattern Recognition, 2001, pp.92 – 99.

[21] J.C. Clarke. Modelling uncertainty: A primer, Technical Report, 2161, University of Oxford, Dept. Engineering Science,1998.

[22] K. Kanatani, Statistical Analysis of focal-length calibration using vanishing points, IEEE Trans. Robotics and Automation, 8 (6) (1992) 767–775.

[23] P.H.S. Torr, Bayesian model estimation and selection for epipolar geometry and generic manifold fitting, International Journal of Computer Vision, 50 (1) (2002) 35-61.

[24] S. M. Seitz and P. Anandan, Implicit representation and scene reconstruction from probability density functions, International Conference on Computer Vision and Pattern Recognition, 1999, pp. 28–34.

[25] M. Barnabei, A.Brini and G-C. Rota, On the exterior calculus of invariant theory, Journal of Algebra, 96 (1985) 120-160.

[26] S. Carlsson, The double algebra: an effective tool for computing invariants in computer vision, in J. Mundy, A. Zisserman and D. Forsyth, editors, Applications of Invariance in Computer Vision, number 825 in LNCS. Springer, 1994.

[27] O. Faugeras and T. Papadopoulo, Grassmann-cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors, In Trans. of the Royal Society A, 365 (1998) 1123–1152.

[28] O. Faugeras and B. Mourrain, On the geometry and algebra of the point and line correspondences between N images, Technical Report 2665, INRIA, 1995.

[29] D. Liebowitz and A. Zisserman, Metric rectification for perspective images of planes, International Conference on Computer Vision and Pattern Recognition, 1998, pp.482–488.

[30] G. Mclean and D. Kotturi, Vanishing point detection by line clustering, IEEE Trans. Pattern Analysis and Machine Intelligence, 17 (11) (1995) 1090-1095.

[31] C. Rother, A new approach for vanishing point detection in architectural environments, in: Proceedings of British Machine Vision Conference, 2000.

[32] S. Barnard, Interpreting perspective images, Artificial Intelligence, 21 (1983) 435-462.

[33] E. Lutton, H. Maitre, and J. Lopez-Krahe, Contribution to the determination of vanishing points using hough transform, IEEE Trans. Pattern Analysis and Machine Intelligence, 16 (4) (1994) 430–438.

[34] G. Schindler and F. Dellaert, Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments, International Conference on Computer Vision and Pattern Recognition, 2004.

[35] J.M. Coughlan and A.L. Yuille. Manhattan world. Neural Computation, 2003.

[36] M. Berg etc. Computational geometry algorithms and applications. ISBN: 3-540-65620-0, published by Springer.

[37] E. Weisstein. Green's theorem, MathWorld, http://mathworld.wolfram.com/ GreensTheorem.htm.

[38] J. Hu, S. You and U. Neumann, Automatic pose recovery for high-quality textures generation. International Conference of Pattern Recognition, HongKong, 2006.

[39] J. Hu, S. You and U. Neumann, Integrating LiDAR, aerial image and ground images for complete urban building modeling, Third International Symposium on 3D Data Processing, Visualization and Transmission, University of North Carolina, Chapel Hill, 2006.

Source table 1

Table 1. Parameter settings.

Parameter	Range	Other parameter settings
1.line orientation angle (degree)	$\theta \in [0.01, 40]$	$fov = 40$ $l = 50$ $\varepsilon = 0.5$ $n = 200$
2.camera field of view (degree)	$fov \in [20, 80]$	$\theta = 1$ $l = 50$ $\varepsilon = 0.5$ $n = 200$
3.image line length (pixel)	$l \in [10, 100]$	$\theta = 10$ $fov = 40$ $\varepsilon = 0.5$ $n = 200$
4.image noise magnitude (pixel)	$\varepsilon \in [0.05, 0.5]$	$\theta = 1$ $fov = 40$ $l = 50$ $n = 200$
5.number of image lines	$n \in [20, 1000]$	$\theta = 5$ $fov = 40$ $l = 50$ $\varepsilon = 0.5$

Source table 2

Table 2. Outdoor real images comparison.

Method	Strong Perspective Image (Orientation error, degree)		Weak Perspective Image (Orientation error, degree)	
	X	Y	X	Y
VH	0.1	0.15	0.3	0.2
ML	0.2	0.3	1.94	0.55

Figure 1

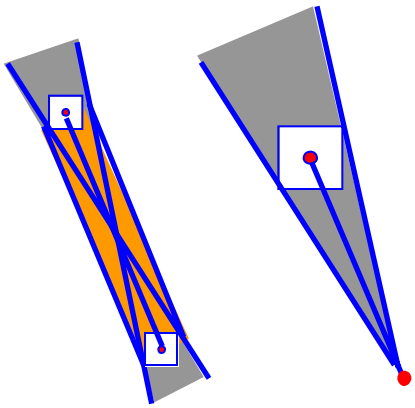


Figure 1. Edge error model.

Figure 2

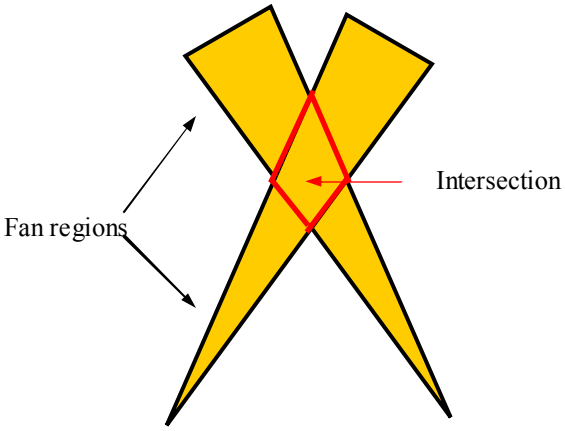


Figure 2. The vanishing hull

Figure 3

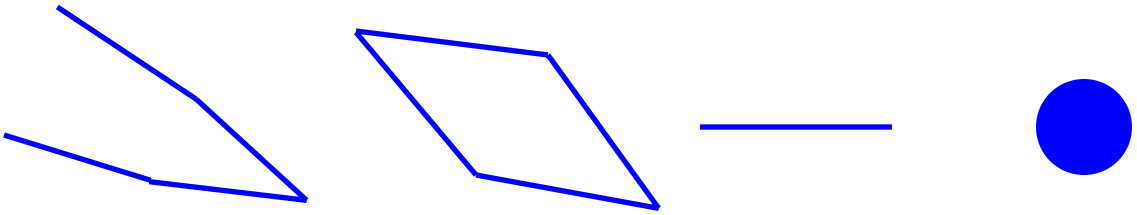


Figure 3. The shape of a vanishing hull. From left to right, open, close, a line and a point.

Figure 4

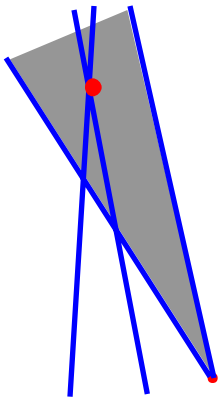


Figure 4. Grouping with an edge error model.

Figure 5

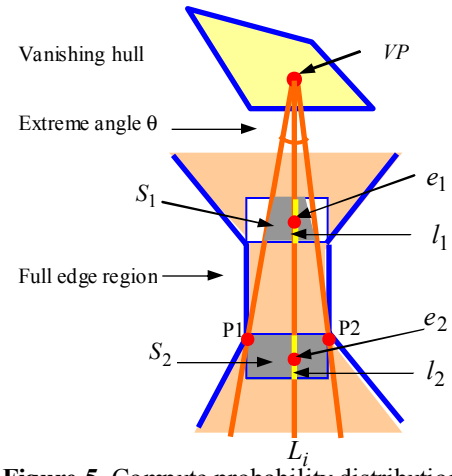


Figure 5. Compute probability distribution for a full edge region.

Figure 6

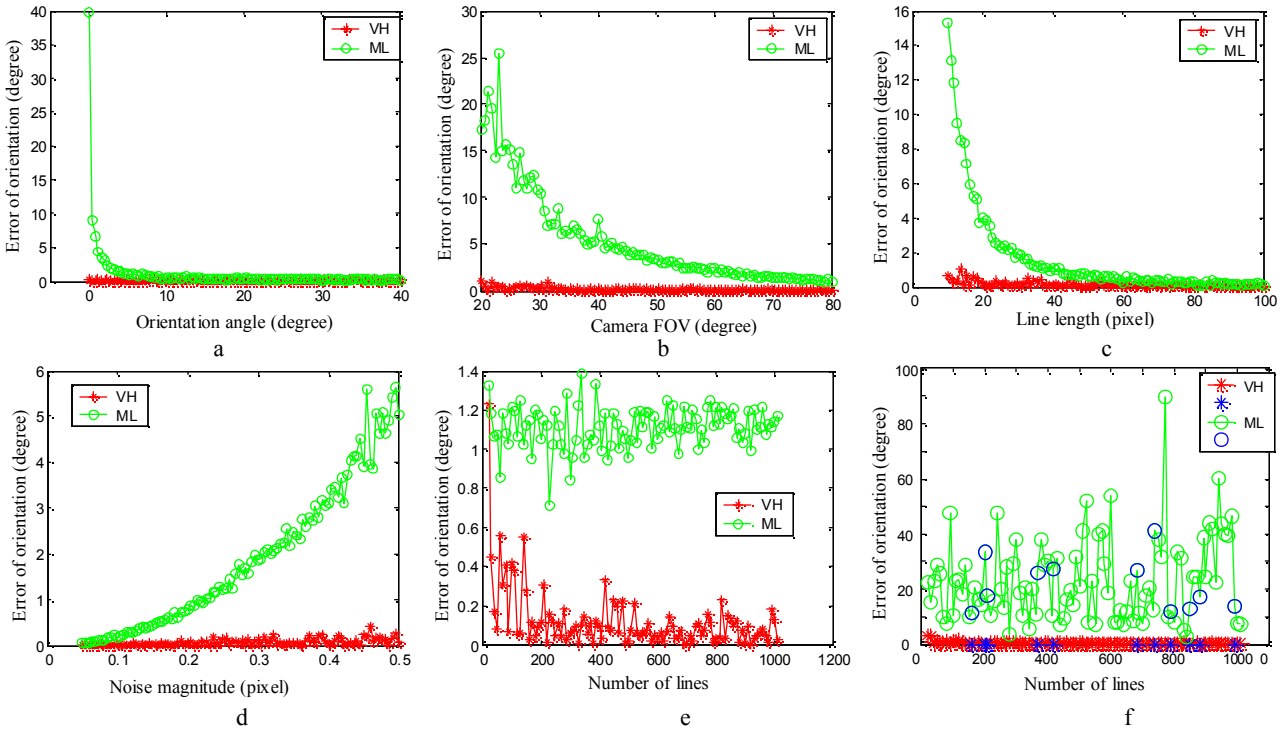


Figure 6. Performance comparison of VH and ML method on synthetic data with random noise.

Figure 7

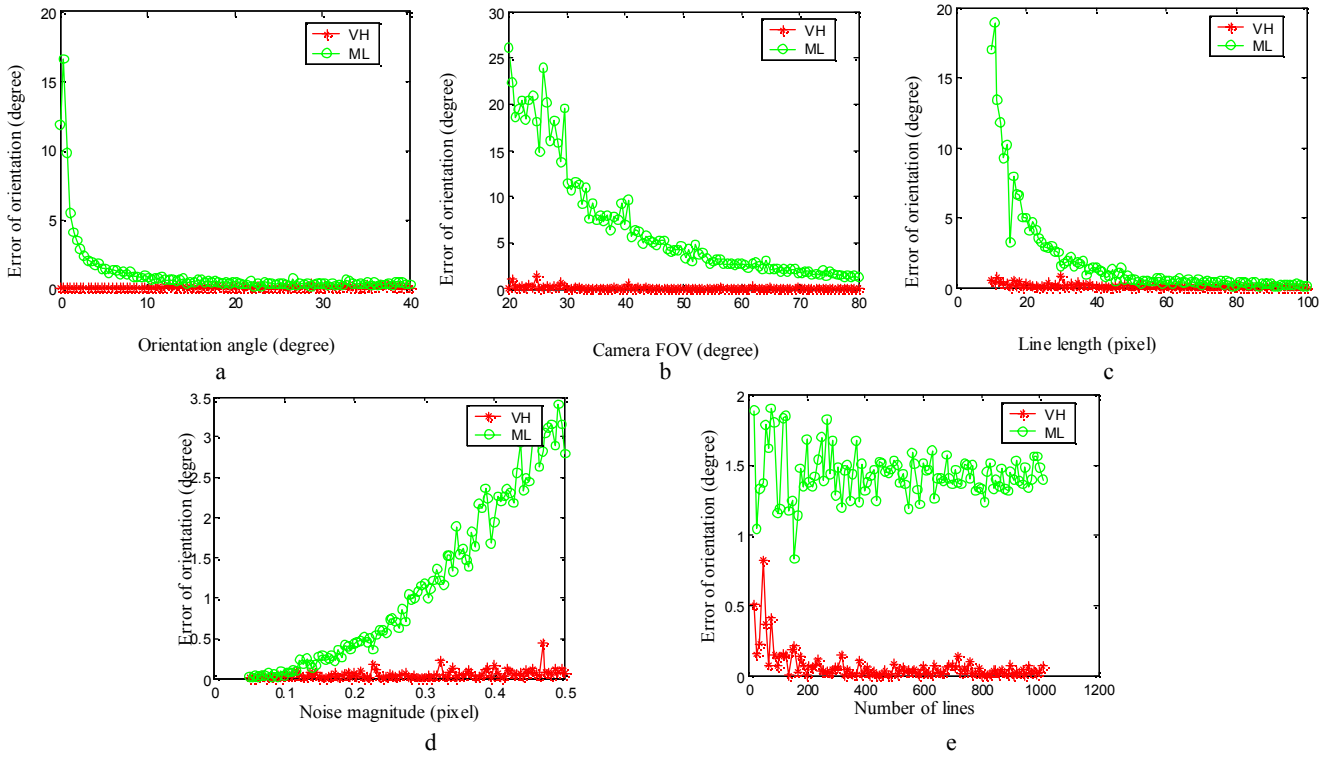


Figure 7. Performance comparison of VH and ML method on synthetic data with Gaussian noise.

Figure 8

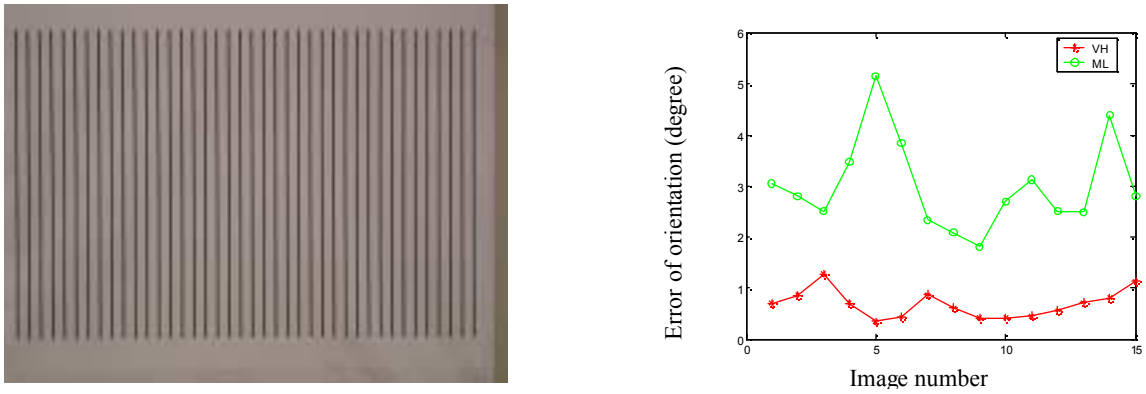
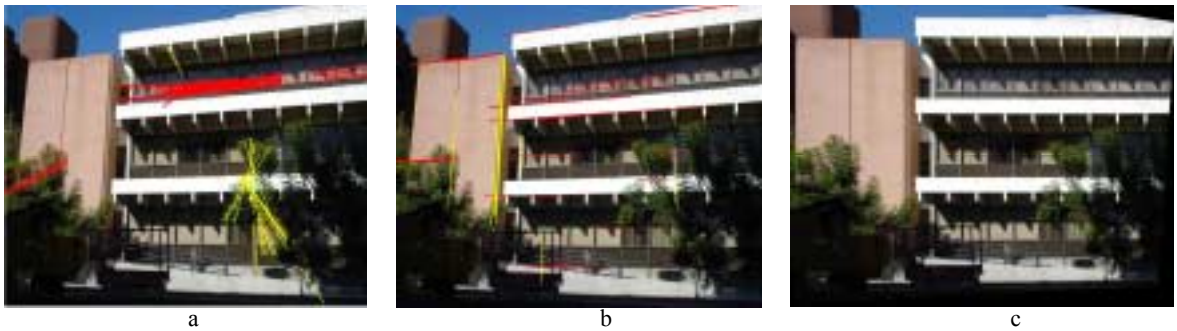


Figure 8. Real data comparison, indoor images. Left: one of the 15 images, right: performance comparison. The error of the VH method (red stars) is much smaller than the ML method (green circles), which shows the advantage of the VH method.

Figure 9



a b c

Figure 9. Our door images: a strong perspective image case. Our vanishing point detection method is robust to trees occlusions and small-scale textures. (a) Clustering result before filtering , (b) the spurious vanishing points are filtered using the heuristic filters, and the image is rectified (c)

Figure 10

Figure 10. Compare rectified image of the VH and ML method for a weak perspective image. Left: ML method, the line that should be horizontal is slanted. Right: VH method, the line is correctly rectified.

Figure 11

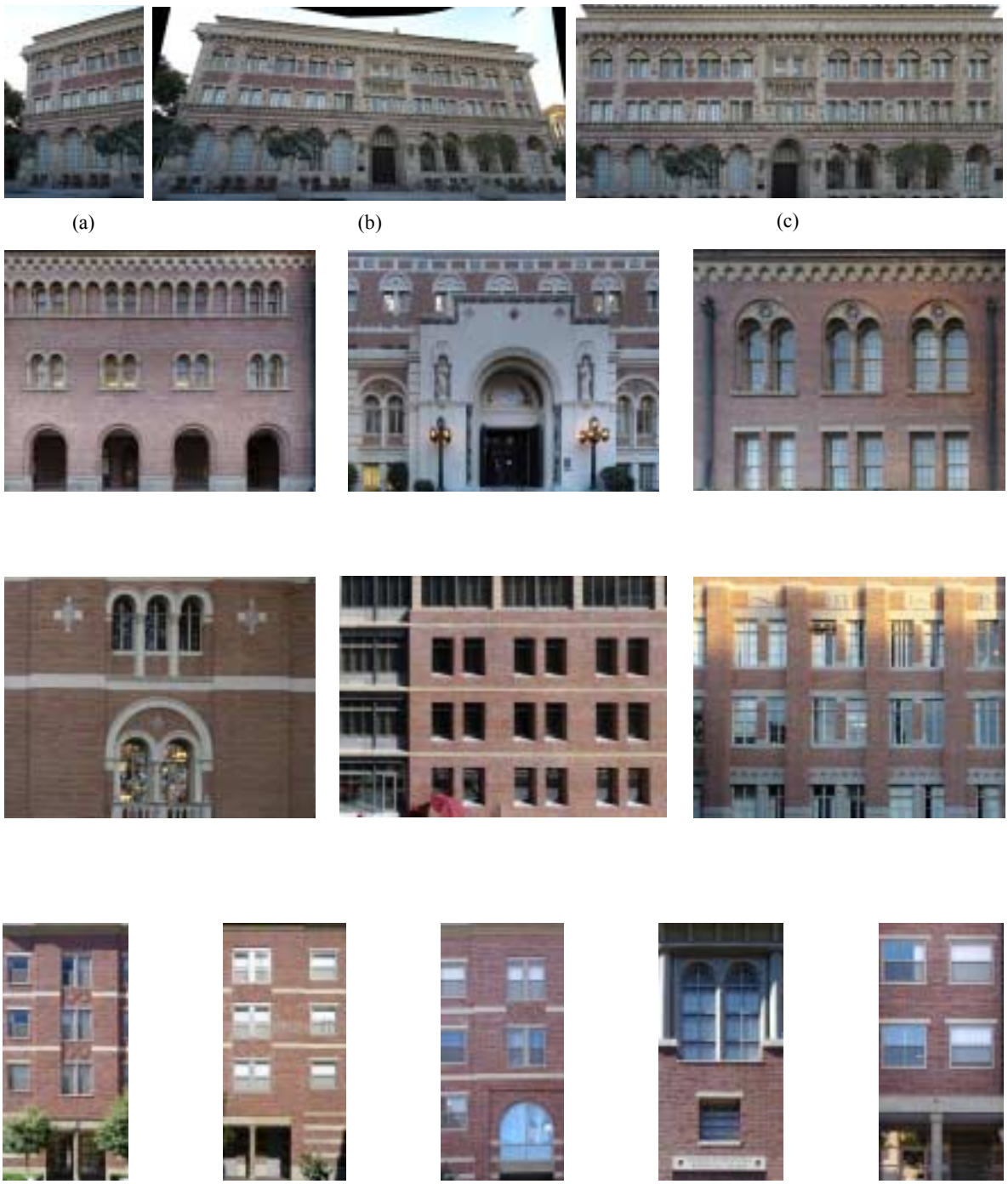


Figure 11. Generate textures. Vanishing hull is used to estimate vanishing points and compute poses, then rectify images to generate textures. One of the original images is shown (a), and the images are stitched as a mosaic (b). The pose is automatically estimated, and the mosaic is rectified to generate a high quality texture image (c). A number of textures rectified using our method are also shown. The rectified textures are visually very accurate.

Figure 12

Figure 12. Rendered images of dozens of textures mapped to 3D models of a university campus.

Figure 13

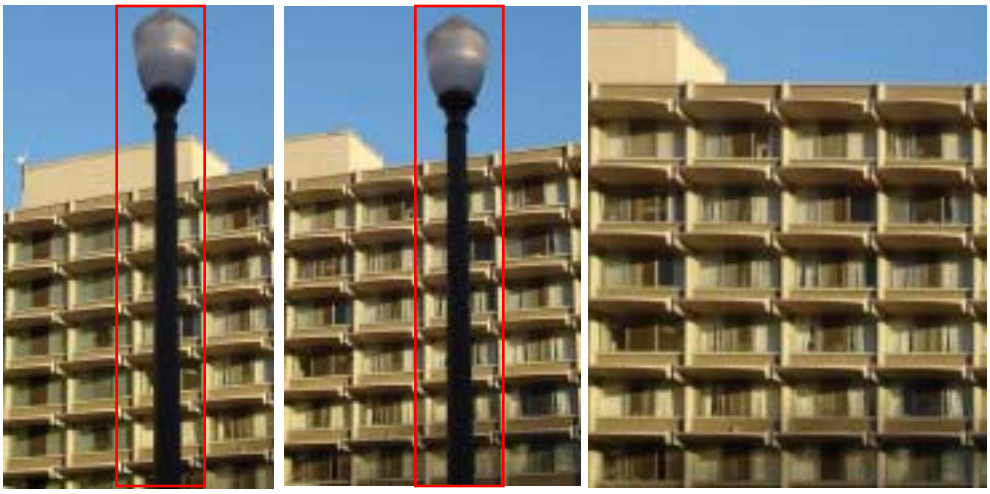


Figure 13. Remove occlusions. Both images ((a) and (b)) are occluded by a pole but at different places. The VH method is used to estimate poses for each image and generate rectified textures with occlusions removed (c).

Figure 14

Figure 14. Rectify an aerial image for hybrid modeling.

Figure 15

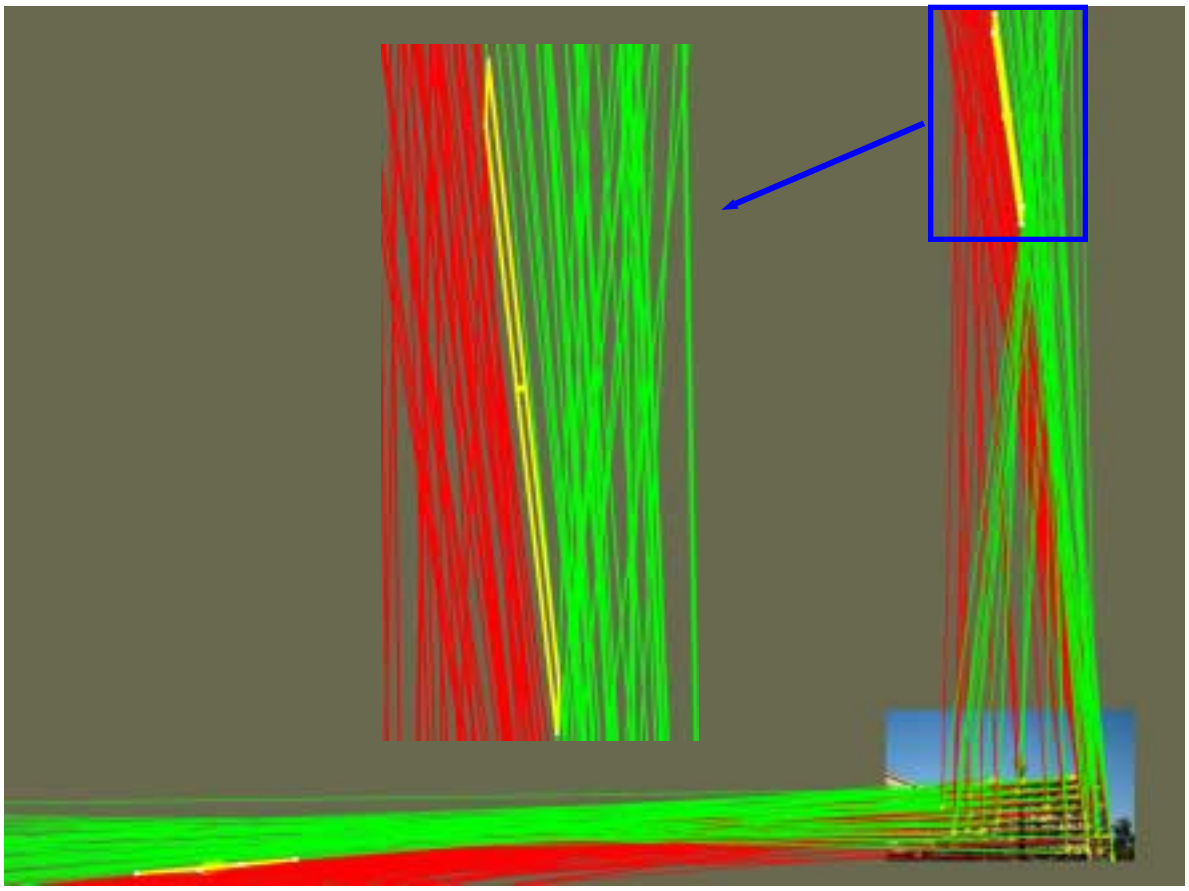


Figure 15. Vanishing hull of a real image. Each line (short yellow lines) in the image forms a fan shape edge region bounded by a red and a green line. These fan regions intersect at a convex polygon (yellow polygon), which is the vanishing hull. An enlarged image of the *y* direction vanishing hull is also shown, where the yellow dot in the center is the centroid of the vanishing hull.