# INTERSECTING 2D LINES: A SIMPLE METHOD FOR DETECTING VANISHING POINTS

*Mariano Tepper and Guillermo Sapiro*

Department of Electrical and Computer Engineering, Duke University
{mariano.tepper,guillermo.sapiro}@duke.edu

**ABSTRACT**

We present a simple and powerful technique for testing with a prescribed precision whether a set of 2D lines meet at a given point. The method is based on a probabilistic framework and has a fundamental geometric interpretation. We use this technique for detecting vanishing points in images. We developed a very simple algorithm that yields state-of-the-art results at a much lower computational cost than its competitors. The presentation of the proposed formulation is complemented with numerous examples.

***Index Terms***— vanishing points; line intersection; a contrario detection.

## 1. INTRODUCTION

In this paper, we develop techniques for addressing an important problem in computer vision: vanishing point detection.

Sets of parallel lines in 3D space are projected into a 2D image as a set of concurrent lines, under the assumption of perspective projection (e.g., with a pin-hole camera). The point in the image plane at which these lines meet is called a *vanishing point*. Vanishing points provide important information to make inferences about the 3D structures of a scene from its 2D (projected) image. Since these projections are non-invertible, concurrence in the image plane does not necessarily imply parallelism in 3D. This said, the counterexamples for this implication are rare in real images, and the problem of finding parallel lines in 3D is reduced in practice to finding vanishing points in the image plane [1, 2, 3, 4]. We are here interested in finding vanishing points when no a priori information is known about the image or the camera with which it was taken.

This task can be reduced to the problem of finding consistently good groups of intersecting lines. In this work, we present a probabilistic and geometric framework for addressing the detection of such intersections.

### 1.1. A probabilistic model for line intersections

We have a set $\mathcal{U}$ of 2D lines that intersect a disk $\mathcal{K}_1$ with radius $r_1$. We define an *intersecting-point (IP) configuration* as the pair $(\mathcal{C}, \mathbf{q})$, where $\mathcal{C} \subseteq \mathcal{U}$ is a set of lines, and $\mathbf{q}$ is a given point ($\mathbf{q}$), such that

$$\forall \ell \in \mathcal{C}, \ell \cap \mathcal{K}_2, \tag{1}$$

where $\mathcal{K}_2$ is the disk centered at $\mathbf{q}$ with some radius $r_2$.

We are interested in computing the probability of an IP configuration under a null random model. Let us assume then that $\mathcal{U}$ is a set of random 2D lines. We consider the event that at least $k = |\mathcal{C}|$ of these $N = |\mathcal{U}|$ lines meet the disk $\mathcal{K}_2$. Under the assumption that all lines are i.i.d., the probability of such an event is $\mathcal{B}(N, k; p)$, where $\mathcal{B}$ is the binomial tail and $p = \Pr(\ell \cap \mathcal{K}_2 \mid \ell \cap \mathcal{K}_1)$. We will later provide details on how to compute $p$.

**Definition 1** ([1])**.** *The IP configuration $(\mathcal{C}, \mathbf{q})$ is said to be $\varepsilon$-meaningful if*

$$\mathrm{NFA}(\mathcal{C}, \mathbf{q}) = N_{tests}\, \mathcal{B}(|\mathcal{U}|, |\mathcal{C}|; p) < \varepsilon, \tag{2}$$

*where $N_{tests}$ is the total number of tested IP configurations. This quantity is usually called number of false alarms (NFA).*

We can be easily prove by linearity of expectation that the expected number of $\varepsilon$-meaningful IP configurations in a finite set of random IP configurations is smaller than $\varepsilon$.

$N_{\text{tests}}$ can also be empirically set by analyzing a training dataset [5], providing a tighter bound for the expectation. In that case, the above lemma holds by construction.

As usual in the a contrario literature [6], we fix $\varepsilon = 1$ once and for all. In this case, we simplify the notation by just referring to *meaningful IP configurations*.

Definition 1 provides a method with a sound probabilistic and geometric meaning for testing whether an IP configuration is likely to happen at random. From the statistical viewpoint the method goes back to multiple hypothesis testing. Following an a contrario reasoning [6], we decide whether the event of interest, i.e., the presence of an IP configuration, has occurred if it has a low probability of occurring by chance in the previously defined random (background) model.

**Organization.** In Section 2, we present a fundamental probabilistic and geometric framework for using this method to detect vanishing points in images. In Section 3 we present some results and, finally, we provide some concluding remarks in Section 4.

## 2. FROM LINE INTERSECTIONS TO VANISHING POINTS

For reasons such as radial distortion and noise, vanishing points cannot be exactly recovered from natural images. In other words, sets of 2D lines that correspond to parallel lines in the 3D scene do not intersect at a single point; however, the intersecting points lie in a relatively small area. Then, we can represent a vanishing point by a small disk around it. Let us explain how to do this in a theoretically sound way.

Let $\mathcal{K}_1$, $\mathcal{K}_2$ be two bounded convex sets in the plane and let $L_1$, $L_2$ be the length of its respective boundaries. Let $L_{\text{out}}$ be the boundary length of the convex hull of $\mathcal{K}_1 \cup \mathcal{K}_2$. We also consider the boundary realized by a closed elastic string drawn about $\mathcal{K}_1$ and $\mathcal{K}_2$, crossing at a point placed between $\mathcal{K}_1$ and $\mathcal{K}_2$. Let $L_{\text{in}}$ denote this boundary's length.

**Theorem 1** ([7, Ch. 3])**.** *The probability that a random line $\ell$ intersecting $\mathcal{K}_1$ also intersects $\mathcal{K}_2$, $\Pr(\ell \cap \mathcal{K}_2 \mid \ell \cap \mathcal{K}_1)$, is*

$$\begin{cases} (L_{in} - L_{out})/L_1 & \text{if} \quad \mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset\,; \\ (L_1 + L_2 - L_{out})/L_1 & \text{otherwise.} \end{cases} \tag{3}$$

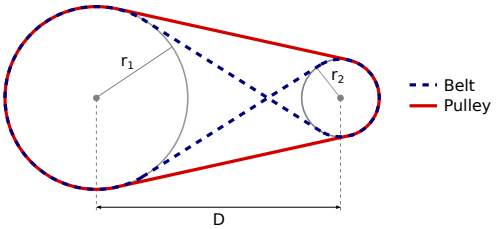



**Fig. 1**: Sketch of the belt and pulley problems.

As pointed out in [1], this theorem provides a very interesting framework for vanishing point detection.

Let $I : \Omega \to \mathbb{R}$ be an image and let $\mathcal{K}_1$ be the disk, with radius $r_1$, whose boundary is the circumcircle of $\Omega$. Suppose also that we have a line detection algorithm that, given an image $I$, yields a set $\mathcal{S}$ of line segments. In this work we use LSD [8] that has shown to yield very stable and robust results. Let $\mathcal{U}$ be the set of lines that contain the line segments in $\mathcal{S}$. Let $\mathbf{p}$, $\mathbf{p}'$ be the endpoints, expressed in homogeneous coordinates, of a segment $s \in \mathcal{S}$. The line $\ell = [\ell_x, \ell_y, \ell_z]^T \in \mathbb{R}^3$ is given by

$$\ell = \mathbf{p} \times \mathbf{p}', \tag{4}$$

$$\ell = (\ell_x^2 + \ell_y^2)^{-1/2}\, \ell. \tag{5}$$

By construction $\forall \ell \in \mathcal{U},\ \ell \cap \mathcal{K}_1$. The distance between a line and a point $\mathbf{q}$ in homogeneous coordinates is given by $\mathrm{dist}(\ell, \mathbf{q}) = \mathbf{q}^T \ell$.

From Theorem 1, we have two cases for computing $p = \Pr\left(\ell \cap \mathcal{K}_2 \mid \ell \cap \mathcal{K}_1\right)$. If $\mathbf{q} \in \mathcal{K}_1$, i.e., the vanishing point is "in the image," the radius $r_2 = r$ is a parameter of the algorithm (in pixels) representing the precision at which we detect vanishing points, and $p = L_2/L_1 = r_2/r_1$. Here we are using the simplifying assumption that $\mathcal{K}_2 \subset \mathcal{K}_1$ if $\mathbf{q} \in \mathcal{K}_1$.

However, if $\mathbf{q} \notin \mathcal{K}_1$ and we want to fairly test all possible IP configurations, we need to change $r_2$ so that $p$ remains fix. In this way, all the possible locations for point $\mathbf{q}$ become equiprobable. We are then interested in solving the following problem: given $p$, $r_1$, the distance $D$ between the centers of $\mathcal{K}_1$ and $\mathcal{K}_2$, and that $\mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset$, we want to compute the radius $r_2$ so that

$$p\, L_1 = L_{\text{in}} - L_{\text{out}}. \tag{6}$$

This problem has a fundamental link with the classical belt and pulley problems. The belt problem requires finding the length $L_{\text{in}}$ of a crossed belt that connects two circular pulleys with radii $r_1$ and $r_2$ whose centers are separated by a distance $D$. Similarly, the pulley problem requires finding the length $L_{\text{out}}$ of a belt that does not cross itself. Both scenarios are depicted in Fig. 1. These problems have closed-form solutions given by

$$L_{\text{in}} = 2\ (r_1 + r_2)\left[\tan(\varphi) + \pi - \varphi\right], \tag{7}$$

$$L_{\text{out}} = 2\,D\,\sin\left(\frac{\theta}{2}\right) + r_1\ (2\pi - \theta) + r_2\,\theta, \tag{8}$$

where $\varphi = \arccos\left(\frac{r_1 + r_2}{D}\right)$ and $\theta = 2\arccos\left(\frac{r_1 - r_2}{D}\right)$.

We are interested in solving the related inverse problem: $L_{\text{in}} - L_{\text{out}}$, $r_1$, and $D$ are given and we want to obtain the radius $r_2$. For this we do a linear approximation around $r_2 = r_0$ and get

$$\begin{aligned}\widetilde{L}_{\text{in}} - \widetilde{L}_{\text{out}} = 2\,P\left[\sqrt{1-R^2} - \sqrt{1-Q^2}\right] + \\ + r_1\left[\arcsin(R) + \arcsin(Q)\right] + \\ + r_2\left[\arcsin(R) - \arcsin(Q)\right],\end{aligned} \tag{9}$$

where $R = \frac{r_0 + r_1}{D}$ and $Q = \frac{r_0 - r_1}{D}$. In practice we set $r_0 = (D - r_1)/2$. From this equation, $r_2$ can be easily obtained. Notice that higher-order approximations (or even Monte Carlo simulations) can be used to augment the precision in the calculation.

**Vanishing points at infinity.** Only parallel lines intersect at infinity. A intersection point $\mathbf{q}$ is at infinity if $q_z = 0$, indicating an orientation in the unit circle located in the plane $z = 0$. Then the distance between a line and such a point $\mathbf{q}$ can then be measured by the angle between the line's orientation and $\mathbf{q}$. Maintaining equiprobability in this case is straightforward: it suffices to set an angular precision $\gamma$ such that $p = \gamma/2\pi$. It is straightforward to see that $\gamma = 2\frac{r}{r_1}\pi$.

**Validating vanishing point detection algorithms.** When the camera internal parameters are known, the absolute conic can be used to measure the accuracy of a vanishing point detection algorithm. However, when such information is not available there is not a clear way for measuring performance. Definition 1, with the NFA computed thanks to the elements we just introduced, provides a geometric and probabilistic framework for evaluating the results of *any* algorithm for detecting vanishing points. Usually, vanishing points evaluation metrics are highly dependent on many technical details. For example, Tardif [4], in order to obtain a distance between an image segment and a vanishing point that is measurable in pixels, backprojects the vanishing point back to the image segment; the obtained distance is then very sensitive to many arbitrary choices for this back-projection.

### 2.1. An algorithm for detecting vanishing points

We now show that the model we proposed for testing IP configurations leads to a simple algorithm for vanishing point detection. The algorithm does not perform any form of random sampling, nor any computationally demanding procedure.

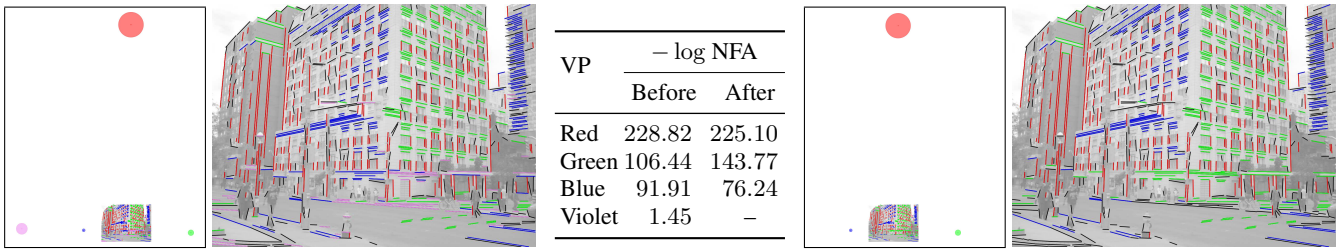
We first define a quality measure for every line. For this, we use the length of the segment from which we computed the line. While more complex measures can be used, e.g., taking into account the contrast along the line, we preferred a simpler approach, emphasizing on the proposed algorithm itself.

The overall procedure is depicted in Algorithm 1. We consider seeds by analyzing the lines in decreasing quality. Lines that belong to a meaningful configuration are discarded from the set $\mathcal{U}$. The meaningful IP configurations already provide very good estimates of the vanishing points (see left side of Fig. 2). We further use a simple fusion algorithm to refine these candidate IP configurations, see Algorithm 3. It operates as a greedy exclusion principle (i.e., a line is only assigned to the most meaningful IP configuration it belongs to), obtaining refined IP configurations (see right side of Fig. 2). We reestimate $\mathbf{q}$ from a set $\mathcal{C}$ using least squares, when $q_z \neq 0$, and by averaging the line orientations, otherwise.

Let us briefly comment on the NFA. The number of tests is given by $N_{\text{tests}} = |\mathcal{U}|(|\mathcal{U}| - 1)/2$. In algorithms 1 and 3, instead of $\mathcal{B}\left(|\mathcal{U}|, |\mathcal{C}|; p\right)$, we use $\mathcal{B}\left(|\mathcal{U}| - 2, |\mathcal{C}| - 2; p\right)$. This is because two seeds are used to compute $\mathbf{q}$, and strictly are not part of $\mathcal{C}$ (although we do keep them in $\mathcal{C}$ for providing a final result). The binomial tail can be efficiently computed using the incomplete beta function [6].

From a worst-case point of view, the algorithm's complexity is quadratic in the size of $\mathcal{U}$. This occurs when the image does not contain vanishing points. However if $\mathcal{U}$ is dominated by sets of concurrent lines, which is the case in many of the images of interest, the complexity becomes almost linear in the size of $\mathcal{U}$.

The vanishing point testing procedure here presented is related to and inspired by [1]. However a deep conceptual difference exists. They perform a tiling of the image plane, and compute NFAs



| VP | − log NFA | |
|---|---|---|
| | Before | After |
| Red | 228.82 | 225.10 |
| Green | 106.44 | 143.77 |
| Blue | 91.91 | 76.24 |
| Violet | 1.45 | – |

**Fig. 2**: Results before fusion (left) and after fusion (right). In both cases, the left image indicate the position of the vanishing points (and the disks $\mathcal{K}_2$) with respect to the image. Colors in the lines and vanishing points indicate meaningful IP configurations. Fusion clearly helps in eliminating some undesired IPs, like the one in violet in this example.

**Algorithm 1:** Vanishing points detection.

**given** $\mathcal{U}, r, \varepsilon$ ; `// In practice we set` $r = 10$
$\mathbf{c} \leftarrow$ image center in homogeneous coordinates;
$r_1 \leftarrow$ radius of the image circumcircle;
$p = r/r_1$; $\mathcal{S} \leftarrow \emptyset$;
**while** $|\mathcal{U}| \geq 2$ **do**
  $\ell_a \leftarrow \underset{\ell \in \mathcal{U}}{\text{argmax}}\ \text{quality}(\ell)$;
  $\ell_b \leftarrow \underset{\ell \in \mathcal{U} \setminus \{\ell_a\}}{\text{argmax}}\ \text{quality}(\ell)$;
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\ell_a, \ell_b\}$;
  $\mathbf{q} \leftarrow \ell_a \times \ell_b$ ; `// 2D line intersection`
  **if** $q_z \neq 0$ **then**
    $\mathbf{q} \leftarrow \mathbf{q}/q_z$;
    compute $r_2$ using Eq. (9);
    **if** $r_2 > r_1$ **then** $\mathbf{q}_z \leftarrow 0$ ;
  **if** $q_z = 0$ **then** $\mathbf{q} \leftarrow \mathbf{q}/\|\mathbf{q}\|_2$ ;
  $\mathcal{C}_{ab} \leftarrow \{\ell_a, \ell_b\}$
  **for** $\ell \in \mathcal{U}$ **do**
    **if** *inlier*$(r, \mathbf{c}, \mathbf{q}, \ell)$ **then** `// see Alg. 2`
      $\mathcal{C}_{ab} \leftarrow \mathcal{C}_{ab} \cup \{\ell\}$;
      reestimate $\mathbf{q}$ from $\mathcal{C}_{ab}$;
  **if** $N_{tests}\, \mathcal{B}(|\mathcal{U}| - 2, |\mathcal{C}_{ab}| - 2; p) > \varepsilon$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\mathcal{C}_{ab}, \mathbf{q})\}$;
    **for** $\ell \in \mathcal{C}_{ab}$ **do** $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\ell\}$ ;
$\mathcal{S} \leftarrow \text{fuse}(\mathcal{S}, |\mathcal{U}|, r, r_1)$ ; `// see Alg. 3`
**return** $\mathcal{S}$

**Algorithm 2:** Inlier check.

**given** $r, \mathbf{c}, \mathbf{q}, \ell$ ;
**if** $q_z \neq 0$ **then**
  **if** $\|\mathbf{q} - \mathbf{c}\|_2 \leq r_1$ **then** $r_2 \leftarrow r$ ;
  **else** compute $r_2$ using Eq. (9) ;
  **return** $(\text{dist}(\ell, \mathbf{q}) < r_2)$;
**else**
  $\alpha = \widehat{\mathbf{q}} - \widehat{\mathbf{e}} + \frac{\pi}{2} \mod \pi$; `//` $\widehat{\mathbf{q}}$ `is the angle of` $\mathbf{q}$
  **return** $(\alpha < 2\frac{r}{r_1}\pi)$;

**Algorithm 3:** IP configuration fusion.

**given** $\mathcal{S}, r, \mathbf{c}$ ;
$\mathcal{V} = \{\ell \mid \exists (\mathcal{C}, \mathbf{q}) \in \mathcal{S}, \ell \in \mathcal{C}\}$;
$\mathcal{S}_F \leftarrow \emptyset$;
**for** $(\mathcal{C}, \mathbf{q}) \in \mathcal{S}$ **do**
  $\mathcal{C}_{\text{new}} \leftarrow \emptyset$;
  **for** $\ell \in \mathcal{V}$ **do**
    **if** *inlier*$(r, \mathbf{c}, \mathbf{q}, \ell)$ **then** $\mathcal{C}_{\text{new}} \leftarrow \mathcal{C}_{\text{new}} \cup \ell$ ;
  $\mathcal{S}_F \leftarrow \mathcal{S}_F \cup \{(\mathcal{C}_{\text{new}}, \mathbf{q})\}$;
**for** $\ell \in \mathcal{V}$ **do**
  $\mathcal{C}_{\text{best}} \leftarrow \underset{(\mathcal{C}, \mathbf{q}) \in \mathcal{S}_F,\ \ell \in \mathcal{C}}{\text{argmin}}\ N_{\text{tests}}\, \mathcal{B}(|\mathcal{U}| - 2, |\mathcal{S}| - 2; p)$;
  **for** $(\mathcal{C}, \mathbf{q}) \in \mathcal{S}_F, \mathcal{C} \neq \mathcal{C}_{best}$ **do** $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\ell\}$ ;
**for** $(\mathcal{C}, \mathbf{q}) \in \mathcal{S}_F$ **do**
  **if** $N_{tests}\, \mathcal{B}(|\mathcal{U}| - 2, |\mathcal{C}| - 2; p) < \varepsilon$ **then**
    reestimate $\mathbf{q}$ from $\mathcal{C}$;
  **else** $\mathcal{S}_F \leftarrow \mathcal{S}_F \setminus \{(\mathcal{C}, \mathbf{q})\}$ ;
**return** $\mathcal{S}_F$;

by counting lines that meet a certain tile. We instead solve the inverse pulley and belt problems and use a simple greedy algorithm for generating the meaningful IP configurations.

## 3. EXPERIMENTAL RESULTS

For evaluating the vanishing point detector, we use the York Urban database [3], which comes with ground truth assignments of image segments to three orthogonal directions in 3D (see examples in the top rows of figs. 3, 4). We compare our results with Tardif's method [4]. Several results are provided in figs. 3 and 4.

The validation procedure is as follows. We first keep the best 3 vanishing points yielded by each method. Since we assume that the camera parameters are not known, Tardif's vanishing points are sorted by decreasing consensus set size and ours are sorted according to the NFA of the IP configurations. Then, for each method, we take each group of ground-truth lines and assign them to the vanishing point that minimizes the NFA.

Some relevant statistics are presented in Table 1. Both methods perform similarly, ours being better at recovering VP1 and VP2, while Tardif's being better at VP3. We would like to emphasize that we are able to obtain state-of-the-art results with an extremely simple (and formal) algorithm, thanks to the various strengths of the proposed testing method.

There are two main situations in which vanishing point detection becomes ill-posed. Some non-manmade objects, such as the tree on the left column of Fig. 4, have structures that challenge the assumption that *only* parallel lines in 3D generate valid IP config-

**Fig. 3**: Examples in which both methods obtain good results. From top to bottom, ground truth, Tardif's method [4], and the proposed method.

| #VPs | $r$ | VP1 | | | | | | VP2 | | | | | | VP3 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Tardif | | | This work | | | Tardif | | | This work | | | Tardif | | | This work | | |
| | | $\rho$ | $\nu$ | $\overline{\text{NFA}}$ | $\rho$ | $\nu$ | $\overline{\text{NFA}}$ | $\rho$ | $\nu$ | $\overline{\text{NFA}}$ | $\rho$ | $\nu$ | $\overline{\text{NFA}}$ | $\rho$ | $\nu$ | $\overline{\text{NFA}}$ | $\rho$ | $\nu$ | $\overline{\text{NFA}}$ |
| 2 or 3 | 5 | 8 | 0.97 | 28.74 | 15 | 0.97 | 31.89 | 15 | 0.99 | 57.68 | 1 | 1.00 | 53.11 | 9 | 0.95 | 30.06 | 13 | 0.96 | 31.34 |
| 2 or 3 | 10 | 5 | 0.95 | 23.38 | 6 | 0.94 | 25.57 | 11 | 0.98 | 44.24 | 1 | 0.99 | 42.59 | 8 | 0.96 | 23.87 | 7 | 0.96 | 24.11 |
| 3 | 5 | 5 | 0.95 | 27.56 | 14 | 0.96 | 30.52 | 10 | 0.99 | 56.15 | 1 | 1.00 | 51.69 | 6 | 0.97 | 29.12 | 12 | 0.98 | 29.62 |
| 3 | 10 | 3 | 0.98 | 22.31 | 5 | 1.00 | 24.33 | 7 | 0.99 | 42.59 | 1 | 1.00 | 40.98 | 5 | 0.96 | 23.05 | 6 | 0.97 | 22.91 |

**Table 1**: Performance comparison of two vanishing point detection methods. Tardif's method always produces at least 3 points, ours might produce less than three if some are non-meaningful. In the first experiment ('2 or 3'), we respect this difference, while in the second experiment ('3'), we only take into account those cases in which ours produces 3 vanishing points. The radius $r$ is the threshold used for testing meaningfulness during validation. $\rho$ denotes the number of cases in which a method obtains a lower NFA than the other. $\nu$ denotes the percentage of meaningful configurations among valid cases. $\overline{\text{NFA}}$ denotes the mean $-log$ NFA among valid cases (although a non-additive quantity, showing the mean is useful).

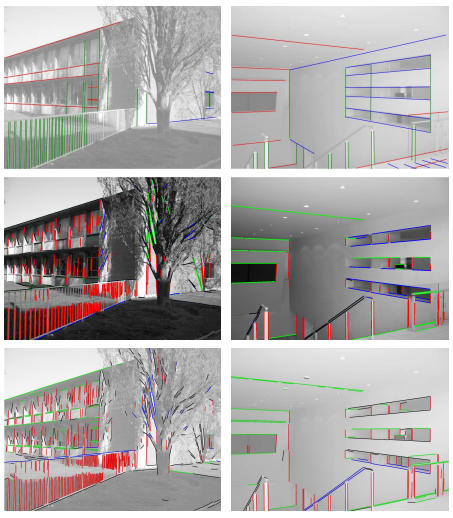


**Fig. 4**: Rare examples where both methods provide less impressive results. From top to bottom, ground truth, Tardif's method, and the proposed method.

urations. The other case is when images are taken from "aberrant" viewpoints, i.e., when strange alignments and/or intersections occur when 3D lines are projected to the image. For example, the tree branches (left column, Fig. 4) form a strange arrangement of line segments. In this cases, both methods yield suboptimal results.

## 4. CONCLUSIONS

We presented a simple but yet powerful technique for testing whether a set of lines meet at a given point with certain precision. Its power comes from the sound probabilistic framework on which it is based. In this work, we used it for detecting and testing vanishing points in images. The proposed technique has a clear and intuitive geometric interpretation, allowing more freedom to intersections that are farther away from the image than to the ones that are closer. We developed a very simple algorithm that yields state-of-the-art results and has, in practice, a linear-time computational cost.

In the future, it might prove useful to consider in the NFA the lengths of the segments supporting a given vanishing point. This additional consideration becomes relevant when a handful of long line segments define a vanishing point.

## 5. REFERENCES


[1] A. Almansa, A. Desolneux, and S. Vamech, "Vanishing point detection without any a priori information," *IEEE TPAMI*, vol. 25, no. 4, pp. 502–507, 2003.

[2] F. Andaló, G. Taubin, and S. Goldenstein, "Vanishing point detection by segment clustering on the projective space," in *ECCV*, 2012.

[3] P. Denis, J. H. Elder, and F. Estrada, "Efficient Edge-Based methods for estimating manhattan frames in urban imagery," in *ECCV*, 2008.

[4] J. P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *ICCV*, 2009.

[5] N. Burrus, T. M. Bernard, and J. M. Jolion, "Image segmentation by a contrario simulation," *Pattern Recognit.*, vol. 42, no. 7, pp. 1520–1532, 2009.

[6] A. Desolneux, L. Moisan, and J. M. Morel, *From Gestalt Theory to Image Analysis*, vol. 34, Springer-Verlag, 2008.

[7] L. Santaló, *Integral Geometry and Geometric Probability*, Cambridge University Press, 1976.

[8] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE TPAMI*, vol. 32, no. 4, pp. 722–732, 2010.