FAST AND ROBUST VANISHING POINT DETECTION ON UN-CALIBRATED IMAGES

Sang Jun Lee and Sung Soo Hwang

Handong Global University

ABSTRACT

This paper presents a novel algorithm for fast and effective vanishing point detection. Once line segments in an input image are detected by LSD algorithm, the proposed method filters out outlier line segments. The remaining line segments are then over-clustered, and each cluster is assigned to 5 different types. According to the assigned type, each cluster is re-merged by applying different criteria, and the re-merged clusters generate hypotheses for vanishing points. Vanishing points are finally detected by utilizing these hypotheses and objective function minimization which reflects orthogonality of vanishing points. The proposed method is accurate because the proposed line over-clustering minimizes erroneous clusters, and type assignment is used for precise re-merging. Furthermore, the proposed method is fast since re-merging is conducted on a cluster level and the objective function is minimized non-iteratively. Experimental results show that the proposed method outperforms the state-of-the-art methods in terms of accuracy and computational cost.

Index Terms— Computer vision, image reconstruction, calibration, projective geometry, clustering method, pose estimation

1. INTRODUCTION

The interest in accurate and fast vanishing point detection has been increased, as it can be used for camera calibration, pose estimation for navigation [15], and single view reconstruction [14]. Vanishing points can be obtained through the intersection of sets of parallel lines corresponding to three axes on the three-dimensional space projected into the image. In order to use sets of parallel lines for vanishing point detection, the axis of structure of the detected object in the image must be orthogonal on the three-dimensional space. Therefore, the key issue of detecting vanishing point is how to find sets of parallel lines which reflects the orthogonality and to cluster them.

For clustering parallel lines, Tardif [10] suggested a line clustering method using J-linkage [8]. J-linkage generates M randomly selected minimal sample set of 2 edges and calculates each consensus set for all detected line segments. Then it compares each consensus set with Jaccard distance and clusters line segments until the consensus sets' distances are all 1s. However, J-linkage has the disadvantages that the line clustering using J-linkage generates erroneous clusters by outlier lines, and detected vanishing points do not reflect orthogonality in three-dimensional space. To reflect the orthogonality, R4L (Random 4 Lines), a RANSAC-based method, was proposed in [2]. R4L utilizes the fact that the degree of freedom of the camera calibration matrix is reduced to one when the principal point is assumed to be located at the center of the image. R4L obtains three vanishing points using two intersection points of randomly selected 4 lines or one intersection point with a polar line. However, R4L sometimes do not estimate the optimal vanishing points on un-calibrated images because the principal point may not be located at the center of an image. Geometry image parsing, proposed by Barinova et al., [5] simultaneously detects line segments, clusters, and vanishing points by minimizing the suggested objective function. The suggested objective function reflects the orthogonality that when the principal point is the center of the image, the zenith line and horizon line are perpendicular to each other. However, this iterative method is computationally expensive.

This paper proposes a new accurate and effective method for fast vanishing point detection. The proposed method consists of line filtering, line clustering, and objective function minimization. The line filtering method filters out the outlier line segments from the line segments detected by LSD algorithm [4] by segment length thresholding. The remaining line segments are over-clustered by J-linkage to reduce type2 [16] errors, and the generated clusters are assigned to a type according to the characteristic of each cluster. Clusters intersecting the same vanishing point are then re-merged by different criteria according to each cluster's type. Each re-merged cluster generates hypotheses for vanishing points, and these are utilized for the proposed objective function minimization. The objective function finally detects vanishing points reflecting orthogonality.

The proposed method shows good performance in terms of accuracy and speed for the following reasons. First, the proposed line filtering algorithm keeps line segments which reflects the structure of scene while reducing outliers, and this reduces the computational cost of the system. Second, by over-clustering and re-merging of line segments, the proposed method minimizes erroneous clusters which enables accurate vanishing point detection. Lastly, the proposed objective function is constructed by considering the orthogonality of the vanishing points, and it is minimized by non-iterative minimization which makes the system fast.

The organization of this paper is as follows. We explain the proposed algorithm and experimental result in Section 2 and Section 3, respectively. And we conclude in Section 4.

2. METHOD

Fig. 1 illustrates the overall structure of the proposed method. Lines detected by the LSD are first filtered to minimize the outlier line, and then the remaining lines are over-clustered. After the over-clustering, each type is assigned to each cluster and re-merging is done according to the type. Finally, vanishing points are detected by minimizing the proposed objective function.

2.1. Line filtering

As shown in Fig. 2 (a), the histogram of the length of the line segments detected by LSD tends to follow f-distribution. Since long line segments represent the structure well, we regard the short line segments as outliers. We specify a threshold δ to remove the line segments whose length is below the δ. We detect the inflection point as threshold as shown in Fig. 2 (b). In order to find the inflection point, we approximate the histogram by a Gaussian kernel, and perform 20-th multinomial regression. Finally, the threshold value is set as the length of maximum value of quadratic differential.

After the thresholding process, the sub-line segments separated from one line must be merged as one line, as proposed in [1]. This is because their intersection points are located at infinity which can cause wrong clustering. Assuming that the focal length is same with the row of the input image, we map segments to the Gaussian sphere [3, 7]. On the Gaussian sphere, a line segment is represented to a great circle, and the great circle is represented by a normal vector. Once an angle between any two normal vectors is less than θ (=1°), the line segments representing the normal vectors are put into strongly connected component as nodes. For the line merging, each strongly connected line segments are merged to a line segment. This process is conducted for all line segments. The angle between two normal vectors is calculated by $\cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2 / \| \mathbf{n}_1 \| \| \mathbf{n}_2 \|)$ where $\mathbf{n}_1, \mathbf{n}_2$ are any two normal vectors, and the typeset bold is 3-vectors.

2.2. Over-clustering

After line filtering, the lines are clustered to sets of parallel lines. For this purpose, Tardif [10] used J-linkage for line clustering. First, a consensus set is constructed as 1 if a line segment s_k is $\mathrm{dist}(s_k, \mathbf{v}_i) < \varepsilon$, or it is set as 0, where $\mathrm{dist}(s_k, \mathbf{v}_i)$ is the image-based consistency measure suggested in [10], $\mathbf{v}_i$ is the i-th hypothesis for vanishing points, and ε is a threshold. Then, lines are merged if Jaccard distance of their consensus sets is not 1. In this method, however, type2 errors [16] may be occurred in a cluster as

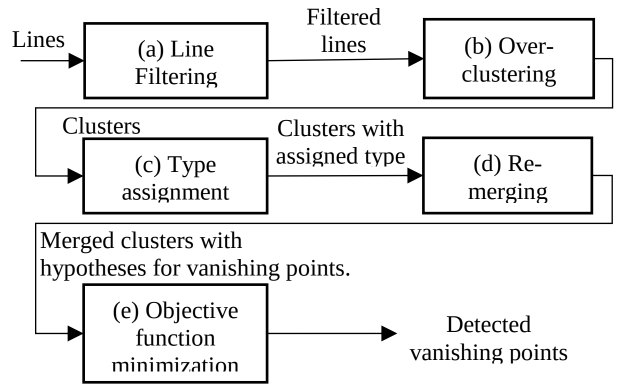
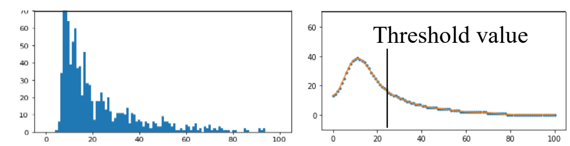


Figure 1. The system over-view of proposed method.



(a) (b)

Figure 2. Graphs for line filtering. (a) A histogram of length of detected lines. (b) An approximated histogram and detected threshold value.

shown in Fig. 3. In Fig. 3, even though the line segment s_k should not be clustered with s_i and s_j, s_k is clustered with s_i and s_j since $\mathrm{dist}(s_k, \mathbf{v}_i) < \varepsilon$. We solve this problem by using small ε enough, and lots of fragmented clusters are generated.

2.3. Type classification

After over-clustering, many clusters are generated with various patterns. Clusters with different patterns cannot be merged in the same way because they have different characteristics. Therefore, as shown in Table 1, different merge methods are applied to each cluster according to its type. The type0 indicates a cluster in which all line segments meet at one vanishing point. The type1 indicates a cluster with one line segment, and the type2 indicates a cluster with two line segments. The type3 is an outlier cluster in which all line segments do not meet at one vanishing point or do not tend to be parallel lines. The type4 is a cluster in which all line segments are parallel.

To classify the type of a cluster in practice, we compute a mean line segment computed for each cluster. The type1 and type2 are decided by the number of line segments in a cluster. The type0 is assigned to the cluster if an intersection point $\mathbf{x}$ of any two line segments in the cluster exist within ε with the mean line segment $\bar{s}$, i.e., $\mathrm{dist}(\bar{s}, \mathbf{x}) < \varepsilon$. The type4 is assigned by analyzing the distribution of intersection points between each line segment and the mean line segment of a cluster. Since intersection points among parallel lines converge to infinity, when the standard deviation of the distribution is sufficiently large (i.e., standard deviation >3000), the type4 is assigned to the cluster. Type3 is assigned

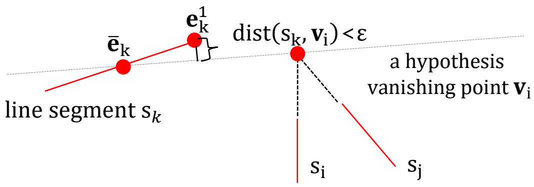


Figure 3. The case of errors that occurred when using J-linkage.

to a cluster if the cluster does not satisfy any conditions.

After type classification, hypothesis for vanishing point of each cluster is calculated according to each type. The type0 and type4 clusters use the intersection point between two arbitrary line segments as hypothesis for vanishing point. The type2 uses just the intersection point of two line segments as hypothesis vanishing point. Neither the type1 cluster nor type3 cluster generates hypothesis for vanishing point.

2.4. Re-merging

After type classification, clusters are re-merged by applying different criteria according to each type. Let $\{\mathbf{v}^{\mathrm{C}}\}$ be the set of all hypotheses for vanishing points of all clusters, $\mathbf{v}_{\mathrm{p}}^{C}$ and $\mathbf{v}_{\mathrm{q}}^{C}$ be hypotheses for vanishing points generated from p-th and q-th clusters, respectively. (i.e., $\mathbf{v}_{\mathrm{p}}^{C}, \mathbf{v}_{\mathrm{q}}^{C} \in \{\mathbf{v}^{\mathrm{C}}\}$, $0 < \mathrm{p} < \mathrm{q} \leq \mathrm{K}$, where K is the number all cluster). And let $\bar{\mathrm{s}}^{C}$ be the mean line segment of a cluster. When p-th and q-th clusters either type0 or type 4, they can be merged if they satisfy the following equation (1).

$$\frac{1}{2}\sqrt{\mathrm{dist}^2\left(\bar{\mathrm{s}}_{\mathrm{p}}^{C}, \mathbf{v}_{\mathrm{q}}^{C}\right) + \mathrm{dist}^2\left(\bar{\mathrm{s}}_{\mathrm{q}}^{C}, \mathbf{v}_{\mathrm{p}}^{C}\right)} < 2 \cdot \varepsilon. \quad (1)$$

In other cases, when p-th and q-th clusters are particularly type0 and type1, or type4 and type1, they are merged if $\mathrm{dist}\left(\bar{\mathrm{s}}_{\mathrm{q}}^{C}, \mathbf{v}_{\mathrm{p}}^{C}\right) < \varepsilon$. When two clusters are merged, the type of the merged cluster follows the type of the first order cluster. The hypothesis of a type2 cluster is directly used for the proposed objective function minimization without re-merging process, since the two lines have an exact one intersection point. A type3 cluster is removed without re-merging, since it is considered as an outlier.

2.5. Objective function minimization

The proposed objective function is defined as the following equation (2):

$$\mathrm{E}(\{\mathbf{v}\}|\mathbf{z}, \{\mathrm{s}\}, \{\mathbf{v}^{\mathrm{M}}\}) = n_c \cdot \left(1 - \frac{\mathrm{L}_{\cap lines}}{\mathrm{L}_{\cup \mathrm{lines}}}\right)^2 + n_{js} \cdot J(\{\mathbf{v}^{\mathrm{M}}\}, \{\mathrm{s}\})^2 + n_{hor} \cdot \phi(\mathbf{u}, \mathbf{z})^2, \quad (2)$$

where $\{\mathrm{s}\}$ is the set of all line segments filtered out, $\{\mathbf{v}^{\mathrm{M}}\}$ is the set of all hypotheses for vanishing points of re-merged clusters. $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}$ and $\mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$ are any two hypotheses in input $\{\mathbf{v}^{\mathrm{M}}\}$,

Table 1. The variety of types

Type	Shape	Type	Shape
Type 0		Type 1	
Type 2		Type 3	
Type 4			

$\mathrm{L}_{\cup \mathrm{lines}}$ is the number of whole line segments in $\{\mathrm{s}\}$, $\mathrm{L}_{\cap lines}$ is the number of segments intersecting $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}$ or $\mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$, $\mathbf{u}$ is a horizontal line, $\mathbf{z}$ is vanishing point for zenith, and the output $\{\mathbf{v}\}$ is the detected three vanishing points including $\mathbf{z}$. n_c, n_{js} and n_{hor} are the weight coefficients of each term. The $\mathrm{L}_{\cap lines}$ uses $\mathrm{dist}\left(\mathrm{s}_n, \mathbf{v}_{\mathrm{p}||\mathrm{q}}^{\mathrm{M}}\right) < \gamma$ to decide whether a line segment s_{n} intersects $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}$ or $\mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$, $\forall \mathrm{s}_{\mathrm{n}} \in \{\mathrm{s}\}$, where the γ is a threshold. $\mathbf{u}$ is $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}} \times \mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$, and $\mathbf{z}$ can be detected where a hypothesis for vanishing point the most lines intersected, having the slop of mean line larger than m (= 10°).

The proposed objective function satisfies three assumptions which reflects orthogonality. First, lines that reflect the structures surely intersect a vanishing point. In other words, a hypothesis the most lines intersected is possible to be a vanishing point. The first term of the function, $\mathrm{L}_{\cap lines}/\mathrm{L}_{\cup \mathrm{lines}}$, means a ratio how many lines intersect $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}$ or $\mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$ among whole lines. By subtracting the ratio from 1, if the $\mathrm{L}_{\cap lines}$ increases, then the return value be small. The second assumption is, similarly to vanishing point detection policy in [2], that the line segments which reflect the structure does not pass through more than two vanishing points. Thus, the second term $J(\{\mathbf{v}^{\mathrm{M}}\}, \{\mathrm{s}\})$ represents the Jaccard similarity between the consensus sets of $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}$ and $\mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$ for $\forall \mathrm{s}_{\mathrm{n}} \in \{\mathrm{s}\}$. The smaller the number of lines crossing the two hypotheses is, the closer the similarity is to 0. Thirdly, as suggested in [5], the horizontal line is perpendicular to the zenith line assuming that the principal point is the center of the image. Thus, the third term $\phi(\mathbf{u}, \mathbf{z})$ is the tangent value between $\mathbf{u}$ and the perpendicular line to the zenith line. The zenith line is a line passing through the $\mathbf{z}$ and the center of the image. Tangent operation returns an exponentially increased value as the angle of two lines increases.

Therefore, given $\mathbf{z}$, $\{\mathrm{s}\}$, and $\{\mathbf{v}^{\mathrm{M}}\}$, $\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}$ and $\mathbf{v}_{\mathrm{q}}^{\mathrm{M}}$ are sequentially put into objective function in the order $0 < \mathrm{p} < \mathrm{q} \leq \mathrm{K}$, K is the number of $\{\mathbf{v}^{\mathrm{M}}\}$. The final vanishing points $\{\mathbf{v}^{*}\}$ are non-iteratively detected as:

$$\{\mathbf{v}^{*}\} = \underset{\mathbf{v}_{\mathrm{p}}^{\mathrm{M}}, \mathbf{v}_{\mathrm{q}}^{\mathrm{M}}}{\mathrm{argmin}}(\mathrm{E}(\{\mathbf{v}\}|\mathbf{z}, \{\mathrm{s}\}, \{\mathbf{v}^{\mathrm{M}}\})). \quad (3)$$

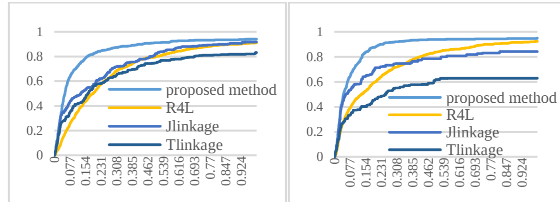
3. EXPERIMENTAL RESULTS

This algorithm is implemented by Python on Window 10 with intel Core i-7 CPU. For the experiment, we used 102 images York Urban Database satisfying the Manhattan world assumption and 103 images Eurasian Cities Database which is the non-Manhattan world. We used R4L, J-linkage, and T-linkage [11] known as all win cases of J-linkage as control group, and they are implemented on python. All methods use line segments detected by LSD with line filtering. For line filtering, the threshold value δ is determined between 20 and 30 automatically, and the number of line segments is reduced from 1000 to less than 500 ones maintaining good structure. The proposed method used $\rho = 10°$, M=100, $\varepsilon = 0.01$ for line clustering and $\gamma = 0.5$, $n_c = 0.2$, $n_{js} = 0.3$, and $n_{hor} = 0.5$ for objective function minimization. The parameters of T-linkage and J-linkage used M = 500 and, as suggested in [11], T-linkage used $\tau = 0.65$, J-linkage, $\varepsilon = \int_0^{5\tau} e^{-\frac{x}{5\tau}} dx\ (\approx 2)$. R4L used K = 500 iteration and $\varepsilon = 0.5$ as presented in [2]. We used horizon estimation error and zenith vanishing point error with cumulative histogram for experimental measurements. Horizon estimation error was proposed in [5], and zenith vanishing point error was measured by the angle between detected zenith vanishing point and ground truth mapped to the Gaussian sphere.

Since all methods are affected by random initial values, we tested dataset 10 times and the experimental results are as follows. The Fig. 4 and Fig. 5 indicate that the proposed method has the best performance in both Eurasian Cities Database and York Urban Database. Since the proposed method finds the vanishing points with the principal point closing to the center of the image instead of fixed center, the detected vanishing points are precise in un-calibrated image. Furthermore, the proposed method was the fastest at min. 0.47 sec and mean 2.46 sec. Since the experiment is done in python, we expect the algorithm to be 0.01 to 0.05 sec in MATLAB code and 0.002 to 0.006 sec in C ++ [13]. The examples of results of the proposed method are as shown in Fig. 7, and the comparison of each method's performance is shown in Fig. 8 where the detected horizontal line is colored by cyan, the ground truth line is yellow, and the clustered lines are colored by red, blue and green.

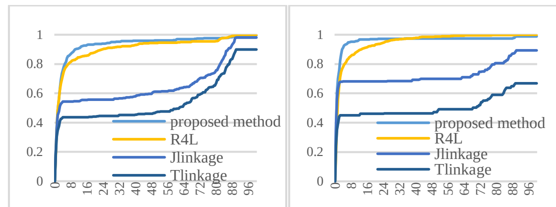
4. CONCLUSION

In this paper, we developed the vanishing point detector with superior performance by over-clustering, type assignment, re-merging and objective function minimization. The proposed method shows that good performance using those processes constructed considering precise and fast approaches reflecting orthogonality for the vanishing points, so it is faster and more accurate than other methods. The proposed method can be used for single view reconstruction or real-time navigation system. For the future, we will continue to develop this algorithm with an optimal solution for more accurate detection using Expectation Maximization method used in [6, 9] and Likelihood maximization used in [2, 12] while considering the speed side.



(a) (b)

Figure 4. Cumulative histograms of horizon estimation error. (a) Eurasian Cities Database. (b) York Urban Database.



(a) (b)

Figure 5. Cumulative histograms of zenith vanishing point error. (a) Eurasian Cities Database. (b) York Urban Database.

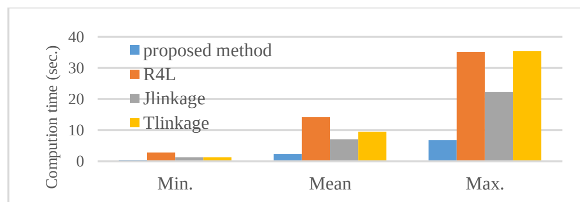


Figure 6. comparison of computation time(sec.)

Figure 7. Examples of results of the proposed method.

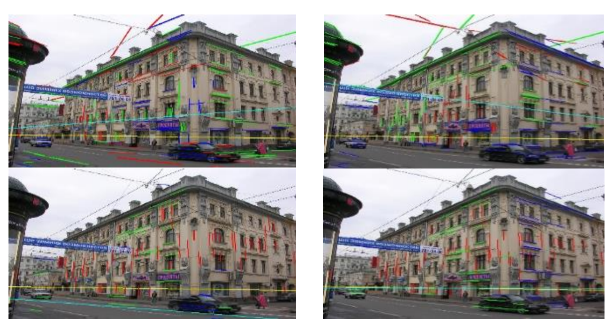


Figure 8. Comparisons of line clustering and horizontal line detection. Top-left is J-linkage, top-right is T-linkage, bottom-left is R4L, and bottom-right is the proposed method.

5. REFERENCES

[1] S.J. Lee and S.S. Hwang, “A Robust Line Merge Method using Gaussian Mapping (in Korean)”, *Image Processing and Image Understanding 2018*, Jeju, Korea, 2018.

[2] H. Wildenauer, and A. Hanbury, "Robust camera self-calibration from monocular images of Manhattan worlds," *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, pp. 2831-2838, 2012.

[3] S.T. Barnard, "Interpreting perspective images." *Artificial intelligence* 21.4 pp. 435-462, 1983.

[4] Von Gioi, R. G., et al. "LSD: a line segment detector." *Image Processing On Line* 2, pp.35-55, 2012

[5] O. Barinova, et al. "Geometric image parsing in man-made environments." *European conference on computer vision*. Springer, Berlin, Heidelberg, pp. 57-70, 2010.

[6] M.E. Antone, and S. Teller, "Automatic recovery of relative camera rotations for urban scenes." *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Vol. 2. IEEE, pp. 282-289, 2000.

[7] J.C. Bazin, et al. "Globally optimal line clustering and vanishing point estimation in manhattan world." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 638-645, 2012.

[8] R. Toldo, and A. Fusiello, "Robust multiple structures estimation with j-linkage." *European conference on computer vision*. Springer, Berlin, Heidelberg, pp. 537-547, 2008.

[9] J. Košecká, Jana, and W. Zhang, "Video compass." *European conference on computer vision*. Springer, Berlin, Heidelberg, pp. 476-490, 2002.

[10] J. P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection." *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 1250-1257, 2009.

[11] L. Magri, and A. Fusiello, "T-linkage: A continuous relaxation of j-linkage for multi-model fitting." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3954-3961, 2014.

[12] Y. Xu, S. Oh, and A. Hoogs. "A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments." *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, pp. 1376-1383, 2013.

[13] S.B. Aruoba, and J. Fernández-Villaverde. "A comparison of programming languages in macroeconomics." *Journal of Economic Dynamics and Control* 58, pp. 265-273, 2015.

[14] R. Hartley, Richard, and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] J.C. Bazin, et al. "Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment." *The International Journal of Robotics Research* 31.1, pp. 63-81, 2012.

[16] Prem S. Mann, *Introductory Statistics 7th Edition*. John Wiley & Sons, 2010.