Real-time Joint Estimation of Camera Orientation and Vanishing Points

Jeong-Kyun Lee and Kuk-Jin Yoon
Computer Vision Laboratory
Gwangju Institute of Science and Technology
{leejk, kjyoon}@gist.ac.kr

Abstract

A widely-used approach for estimating camera orientation is to use points at infinity, i.e., vanishing points (VPs). By enforcing the orthogonal constraint between the VPs, called the Manhattan world constraint, a drift-free camera orientation estimation can be achieved. However, in practical applications this approach suffers from many spurious parallel line segments or does not perform in non-Manhattan world scenes. To overcome these limitations, we propose a novel method that jointly estimates the VPs and camera orientation based on sequential Bayesian filtering. The proposed method does not require the Manhattan world assumption, and can perform a highly accurate estimation of camera orientation in real time. In addition, in order to enhance the robustness of the joint estimation, we propose a feature management technique that removes false positives of line clusters and classifies newly detected lines. We demonstrate the superiority of the proposed method through an extensive evaluation using synthetic and real datasets and comparison with other state-of-the-art methods.

1. Introduction

The projections of parallel lines onto an image plane intersect at a point called a vanishing point (VP). Because of the advantages of its special geometric attributes, the VP has been extensively studied in computer vision communities and employed in many applications, such as camera calibration [15, 6] and rotation estimation [3, 20, 19, 16]. In particular, the estimation of camera orientation using VPs is applied to 3D scene reconstruction [20, 19] and vehicle control [16, 11] because a VP is a translation-invariant feature and therefore the rotation estimation can be more accurate by using VPs.

In previous researches [3, 20, 19], rotation estimation using VPs has been studied for elaborate 3D reconstruction of urban scenes from multiple wide-baseline images. In the case of urban scenes, the so-called Manhattan world constraint, where a triplet of mutually orthogonal VPs commonly appears, is imposed on the 3D reconstruction because the constraint allows VPs to be matched more easily in wide-baseline images and aligned more precisely. Thus, the methods produce highly accurate rotation estimates in cooperation with optimization techniques.

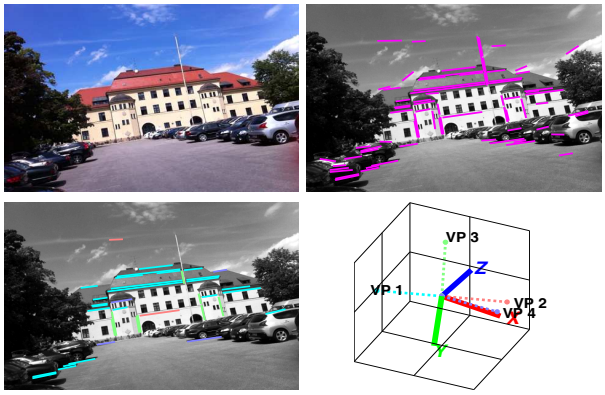


Figure 1. In a general scene (top left), given the line segments detected in the image sequence (top right), the proposed method clusters parallel lines (bottom left) and jointly estimates the camera orientation and VPs (bottom right).

In recent years, the computation of rotation from VPs has been implemented for smartphone applications and unmanned aerial vehicles (UAV), and therefore noise-robust and real-time processing in sequential images is required. In [8, 4, 5], methods for achieving robust and real-time rotation estimation in an image sequence were proposed. A key point of the methods is to incorporate the Manhattan world constraint into the rotation estimation. This allows us to quickly find the VP correspondences in a small-baseline image sequence and improve the robustness of the rotation estimation even when noisy line segments are extracted. However, some difficult problems still remain. The previous methods use accumulation- or sample consensus-based VP detection techniques, and therefore many dominant parallel lines corresponding to VPs should appear and be observed on an image. Unfortunately, in the real-world scenes, a cluster of parallel lines often includes many spurious parallel lines on the image, or the scene may be a non-

Manhattan world scene. Therefore, the rotation estimates by the existing methods are prone to be inaccurate and unstable in practical applications in general scenes.

In this paper, we propose a novel method that jointly estimates camera orientation and VPs based on nonlinear Bayesian filtering as shown in Fig. 1. We first develop the system models for the joint estimation in order to enable the proposed method not to require the Manhattan world constraint and but to perform robustly using a few VPs that are not necessarily orthogonal to each other. This also improves the accuracy of the estimation. In addition, we propose a feature management technique, which detects and removes parallel line segments and VPs, to enhance the robustness against spurious lines and VPs. Consequently, the proposed method provides a highly accurate and robust performance in general scenes.

The paper is organized as follows. We review the literature on VPs and rotation estimation in Section 2 and represent a brief description of background information related to this work in Section 3. We then describe the details of the proposed method in Section 4 and demonstrate its superiority with experimental results using synthetic and real data in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

A VP is independent of the camera's position, and therefore it has been widely used for estimating the rotation of the camera more accurately as in [3, 20, 19, 8, 4, 5]. Most previous researches focused on VP detection and matching among multiple images, which are the main issues in the rotation estimation using VPs.

Some camera orientation estimation methods [3, 20, 19] were developed for more accurately reconstructing a 3D scene using multiple *wide-baseline images*. In the case of wide-baseline images, it is very difficult to match VPs as well as to distinguish spurious VPs. For this reason, approximations of camera orientation were initially given in [3, 20]. In [3], VPs were estimated based on the expectation maximization (EM) algorithm in Hough space and VP correspondences were determined through an iterative scheme. Unlike in [3], where the accumulation-based method that clusters VPs in discretized space was used, in [20] VPs were estimated by using a multiple RANSAC-based method, and three mutually orthogonal VPs were found in each view and then matched. The VP correspondences could be found more easily by the Manhattan world assumption. In [19], VPs were matched using a color histogram-based line descriptor. This method is able to estimate camera orientation without the initial rotation approximations. All the methods presented in [3, 20, 19] commonly optimized the rotation estimate in the final step for elaborate 3D reconstruction, which is computationally expensive.

Recently, methods for real-time performance in sequential images, *i.e.*, *small-baseline images*, were proposed. In an image sequence, VP correspondences can be found more easily than in wide-baseline images since matching a VP is restricted to a narrow range. The simplest method was to detect VPs, which are not necessary to be orthogonal to each other, by using VP extraction methods in [22, 25, 14] and then match VPs by selecting the nearest VP in the next frame. However, the VP matching method provided unreliable matches because temporally unstable and inconsistent VP extraction was caused by noisy or spurious parallel line segments. Hence, several methods in [8, 4, 5] solved the problem by enforcing the Manhattan world constraint as well. The method in [8] found triplets of orthogonal VPs using the RANSAC-based method and then matched the triplets in consecutive frames by computing an Euclidean distance between VPs. However, the method was not robust in the presence of noisy or spurious parallel lines. In [4], a top-bottom approach was proposed based on an exhaustive search that found a maximal consensus set of mutually orthogonal parallel lines for given rotational hypotheses. Accordingly, this method was more robust than that proposed in [8] but did not guarantee a globally optimal solution of the maximum consensus set. On the other hand, in [5] the author adopted a branch-and-bound (BnB) strategy, which guarantees the optimality of the solution. Nevertheless, these methods still have significant limitations: 1) finding VP correspondences is based on the Manhattan world constraint, and 2) the rotation estimate is prone to be inaccurate and unstable in the presence of many noisy or spurious line measurements. In this paper, we propose a novel method for accurately and rapidly computing camera orientation without the Manhattan world constraint by estimating camera rotation and VPs jointly.

3. Background knowledge

3.1. Gaussian sphere

The Gaussian sphere is a unit sphere centered on the principle point, *i.e.*, the center of projection, in the pinhole camera model. As shown in Fig. 2, on the space, a line on the image plane is represented by a great circle, which is the intersection of the unit sphere and a plane defined by the line and the center of projection. All the great circles of parallel lines intersect at a point on the unit sphere. A direction from the center of projection to the intersection point eventually becomes a vanishing direction (VD). The VD is orthogonal to all the normals of the great circles.

3.2. Rotational dependence of a vanishing point

Given a VD on 3D space $\mathbf{d} \in \mathbb{R}^3$, the VD on homogeneous coordinates $\mathbf{D} \in \mathbb{P}^3$ is represented as

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}^{\mathrm{T}} & 0 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} X & Y & Z & 0 \end{bmatrix}^{\mathrm{T}}. \quad (1)$$

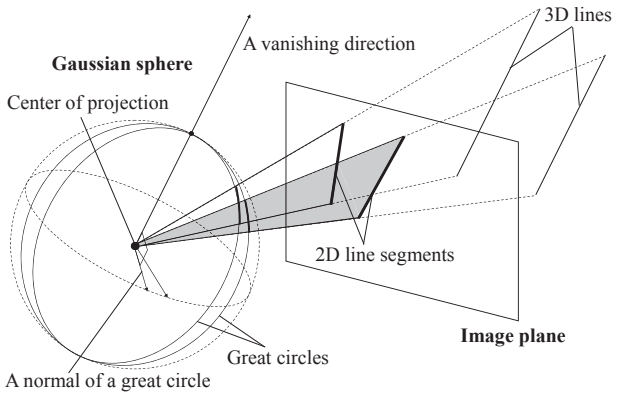


Figure 2. The lines in 3D space are projected onto the line segments on the image plane. A plane composed of one line segment and the center of projection crosses over the Gaussian sphere in which a great circle is formed. The normals of the great circles of the parallel lines in 3D space are orthogonal to a vanishing direction estimated from a set of the parallel lines.

The vector $\mathbf{D}$ can be transformed into $\mathbf{D}'$ by a 4×4 matrix representing rotational and translational transformations in Euclidean 3D space as

$$\mathbf{D}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{D} = \begin{bmatrix} \mathbf{Rd} \\ 0 \end{bmatrix}, \tag{2}$$

where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ the translation vector. From Eq. (2), the transformed VD $\mathbf{d}'$ equals $\mathbf{Rd}$. This means the Euclidean 3D transformation of a VD is influenced only by rotation. Since a VP is a projection of the VD, the VP also has the same property. Given a 3×4 camera projection matrix $\mathbf{P}$ that consists of a camera calibration matrix $\mathbf{K}$, a rotation matrix $\mathbf{R}_{CW}$, and a translation matrix $\mathbf{t}_{CW}$ relative to the camera, a VP $\mathbf{p} \in \mathbb{P}^2$ for a VD $\mathbf{D}$ is

$$\mathbf{p} = \mathbf{K} \begin{bmatrix} \mathbf{R}_{CW} \mid \mathbf{t}_{CW} \end{bmatrix} \mathbf{D} = \mathbf{K}\mathbf{R}_{CW}\mathbf{d}, \tag{3}$$

which also shows that the VP depends on the camera rotation only.

4. Joint estimation of camera orientation and vanishing points

The main idea of the proposed method is to jointly estimate camera orientation and VPs by utilizing the knowledge that a line, a VP, and the camera orientation are geometrically correlated as described in Section 3. That is, given the parallel line segments observed in the image, the proposed method estimates the camera orientation and VPs jointly by using the correlation. This means that the method does not match orthogonal VPs across images to find the rotation, but rather tracks only the parallel line segments. Therefore, it is not necessary to enforce the Manhattan world constraint. In addition, our system model design takes into consideration the uncertainties of measurement and camera motion. The proposed method consequently improves the accuracy of the estimation, despite of measurement noise.

For the joint estimation, we design a nonlinear Bayesian filtering framework based on the geometric property, *i.e.*, the rotational dependence, of a VP. We consider two geometric properties: a VD is orthogonal to the normal of a great circle corresponding to a line measurement, and VPs are dependent only on the rotation of the camera. These two properties lead to a correlation between a line, a VP, and the camera orientation. In Section 4.1, the design of a camera motion model and measurement model in which this correlation is implemented is described[1]. In this work, we use the extended Kalman filter (EKF) system [18] for nonlinear Bayesian filtering. It is noteworthy that a non-iterative scheme of the EKF is computationally efficient. Thus, we can expect the system to perform rapidly. Moreover, additional proposed steps, such as the RANSAC-based outlier rejection described in Section 4.3 and the feature management described in Section 4.4, improve the accuracy and robustness of the proposed method. Figure 3 shows the proposed joint estimation system.

4.1. System modeling

The proposed system estimates a state of the camera and VPs based on Bayesian filtering. The state vector $\mathbf{x}$ is defined by $\mathbf{x} = [\mathbf{x}_v^{\mathrm{T}},\ \mathbf{y}_1^{\mathrm{T}},\ \mathbf{y}_2^{\mathrm{T}},\ \cdots]^{\mathrm{T}}$, where $\mathbf{x}_v$ is a camera state vector and $\mathbf{y}_i$ is a VD vector. The camera state vector is composed of a quaternion for the camera orientation in world coordinates $\mathbf{q}_{WC}$ and an angular velocity relative to the camera ω_C and defined by $\mathbf{x}_v = [\mathbf{q}_{WC}^{\mathrm{T}},\ \omega_C^{\mathrm{T}}]^{\mathrm{T}}$. A VD vector $\mathbf{y}_i$ is defined as $\mathbf{y}_i = [\theta_i, \phi_i]^{\mathrm{T}}$. The two elements θ_i and ϕ_i, which represent angles between each axis, are used to express a direction. This representation is sufficient to denote a VD in world coordinates and efficient for estimating a VD since it represents the directional characteristic of a VD more effectively than a directional vector in Cartesian coordinates does; that is, a covariance of a VD estimated in spherical coordinates will be more informative than the covariance computed in Cartesian coordinates when the state is updated in the nonlinear stochastic filtering. A newly detected VD vector $\mathbf{y}_{new}$ is additionally augmented after the last VD vector of the state vector.

4.1.1 System model

The estimation based on the Bayesian filtering uses a motion model, which is necessary for predicting the next state. Since we aim only at estimating the camera orientation in an image sequence, we consider a constant angular velocity model as the motion model for the system. It may be a less limited and more reasonable assumption than the Manhattan world assumption in image sequence processing. Our

[1]More detailed derivations are provided in the supplementary material.

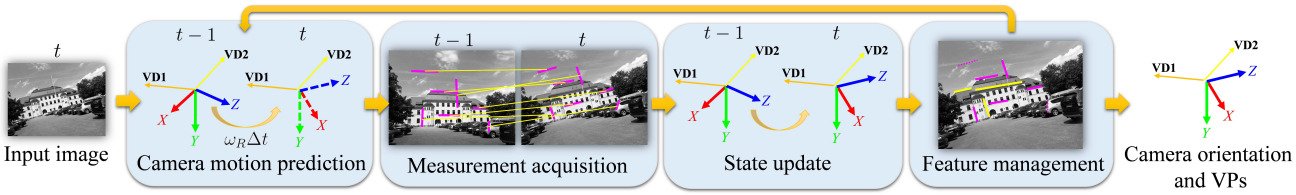


Figure 3. A flowchart of the proposed joint estimation of camera orientation and VPs

system model function $\mathbf{f}_v$ is then defined as

$$\mathbf{f}_v = \begin{bmatrix} \mathbf{q}_{WC}^{new} \\ \omega_C^{new} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{WC}^{old}\mathbf{q}((\omega_C^{old} + \mathbf{\Omega})\Delta t) \\ \omega_C^{old} + \mathbf{\Omega} \end{bmatrix}, \quad (4)$$

where $\mathbf{\Omega}$ is a noise of the angular velocity and $\mathbf{q}(\theta)$ a quaternion function for an angle θ.

4.1.2 Measurement model

To design a measurement model, we exploit the property that a VD is orthogonal to a normal of a great circle of a line parallel to the VD as in Fig. 4. A measurement model h_{ij} for the i-th VD of the state vector and the j-th line feature is defined by

$$h_{ij} = \mathbf{d}_i^{\mathrm{T}}\mathbf{R}(\mathbf{q}_{WC})\mathbf{n}_{ij}, \quad (5)$$

where $\mathbf{d}_i$ is the i-th unit VD vector transformed from spherical coordinates to Cartesian coordinates, $\mathbf{R}(\mathbf{q})$ the transformation of a quaternion $\mathbf{q}$ into a 3×3 rotational matrix, and $\mathbf{n}_{ij}$ the unit normal vector of the great circle defined by the j-th line feature parallel to the i-th VD in the camera coordinates. The normal $\mathbf{n}_{ij}$ is computed from a plane passing through the camera center and both endpoints of the j-th line on the image plane. A value of the measurement model h_{ij} itself is considered as a measurement residual, which is used to update the current state. All the line measurements parallel to the i-th VD are used to update the camera orientation and the i-th VD.

4.2. Initialization

At the beginning of the estimation, it is necessary to detect VPs and then initialize a state vector and its covariance. First, we employ the multiple RANSAC-based VP detector [17] in order to cluster the lines extracted by a line segment detector. We test two line segment detectors, LSD [23] and EDLines [2] since they offer a trade-off between accuracy and computational efficiency. We will show the performance according to the detectors in Section 5. From among many clusters taken from the VP detector, we select a few clusters that the numbers of the parallel lines are larger than a threshold (in our experiments, 6) since they may include false positive clusters. Then, the clustered lines are used to approximately compute the VDs. The computation is formulated as a problem that minimizes the inner products of an unknown VD vector and the normals of the great circles

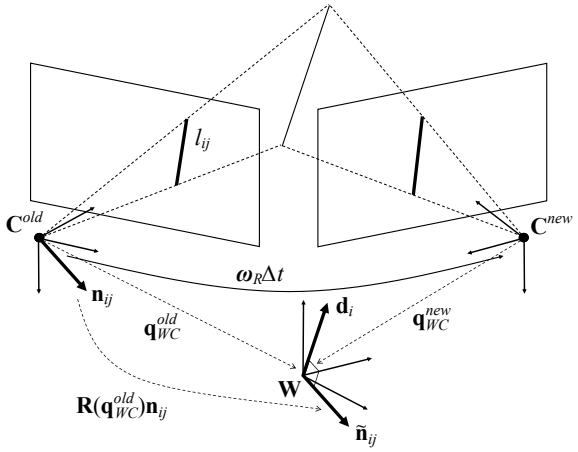


Figure 4. When the camera moves from $\mathbf{C}^{old}$ to $\mathbf{C}^{new}$, the camera orientation $\mathbf{q}_{WC}^{old}$ is changed to $\mathbf{q}_{WC}^{new}$ according to the rotation $\omega_C\Delta t$. A normal $\mathbf{n}_{ij}$ corresponding to a line l_{ij} is orthogonal to a vanishing direction $\mathbf{d}_i$ of the line l_{ij}.

defined from the clustered lines. We solve the minimization problem using singular value decomposition, and then the approximately estimated VDs are augmented in the state vector as initial values. In addition, the sets of the clustered lines are dealt with in the feature management step, which is described in Section 4.4.

To initialize a covariance of a new VD $\mathbf{y}_{new}$, we consider uncertainties of the VD and the camera orientation since the initial estimate of the new VD contains noise of the current camera orientation. The covariance $\mathbf{P}_{\mathbf{y}_{new}}$ is defined as

$$\mathbf{P}_{\mathbf{y}_{new}} = \frac{\partial \mathbf{y}_{new}}{\partial \mathbf{x}_v}\mathbf{P}_{\mathbf{x_v}}\frac{\partial \mathbf{y}_{new}}{\partial \mathbf{x}_v}^{\mathrm{T}} + \widetilde{\mathbf{P}}_{\mathbf{y_{new}}}, \quad (6)$$

where $\frac{\partial \mathbf{y}_{new}}{\partial \mathbf{x}_v}$ is a jacobian of the new VD for the camera state, $\mathbf{P}_{\mathbf{x_v}}$ a covariance of the current camera state, and $\widetilde{\mathbf{P}}_{\mathbf{y_{new}}}$ a noise of the estimate of the new VD. The covariance $\widetilde{\mathbf{P}}_{\mathbf{y_{new}}}$ is initially set along the environment.

4.3. Measurement acquisition

In the proposed method, measurements are obtained by tracking line features at each frame. The tracking of line segments has been addressed in many previous studies. Here, we consider two methods: a simplified and a descriptor-based method.

First, we consider a simple method to quickly track line segments for reducing the computational load of the system. We detect line segments in a current image using the line segment detector. Then, the length of each line and the gradient at the center point of each line are computed. We use the noise robust gradient operator [12] for computing the gradient. It provides a more reliable description for a line. The line features are updated by matching the detected line segments. If the rate of the lengths is between $1/l_{th}$ and l_{th}, the difference in the gradient magnitudes is below m_{th}, and the angle between the gradient directions of two line segments is below θ_{th}, then the line segment is regarded as a correspondence candidate. From the set of the candidates, we select one correspondence with the minimum difference in gradient magnitudes. In the experiments, the parameters l_{th}, m_{th}, and θ_{th} are set to 1.5, 30, and 15, respectively.

We also consider the descriptor-based method, which produces a better matching performance particularly in a complex scene although it is slower than the above simple method. In this method, a line segment is accepted as a candidate in a manner similar to that in the above simple method. Here, the distance of the descriptors should be below $\bar{m}_{th}$ instead of the magnitude of the gradient. We then select one correspondence with the minimum distance of the descriptors. A modified LBD descriptor [26] is used in our method since the LBD descriptor is simpler and faster than other descriptors [24, 9] and has high distinctiveness. The modified descriptor is computed by using sample points equally spaced in the line support region. This is computationally efficient and does not degrade the matching performance. We set the parameters l_{th}, $\bar{m}_{th}$, and θ_{th} to 4, 0.3, and 25, respectively, because of the high distinctiveness of the descriptor. The number of bands and the width of a band for the LBD descriptor are set to 5 and 7, respectively.

In addition, we perform a RANSAC-based outlier rejection inspired by [7]. In this work, three lines are sampled to generate a hypothesis. This offers a more accurate and robust performance while the resulting increase in computational time is small.

4.4. Feature management

VP estimates and line features often become unreliable because of incorrect line tracking or VP detection. This is fatal to the rotation estimation in the joint estimation method. We propose some techniques for managing the VPs and line features to achieve a noise-robust and long-term estimation.

We employ the rotational dependence for detecting new line features. We first calculate the normal of the great circle for all the line segments newly extracted in the current frame. Then, the line segment is accepted as a line feature candidate if an absolute value of an inner product of the VD and the normal is less than a threshold α_{th}. If the candidate is tracked during a few frames or a rotation by a few degrees, the candidate is registered as a new line feature in the class of the VD. Here, we empirically set α_{th} to 0.02. If a line feature is not observed during a few frames or the normal of its great circle is regarded as not being orthogonal to its VD, we remove the line feature in the class.

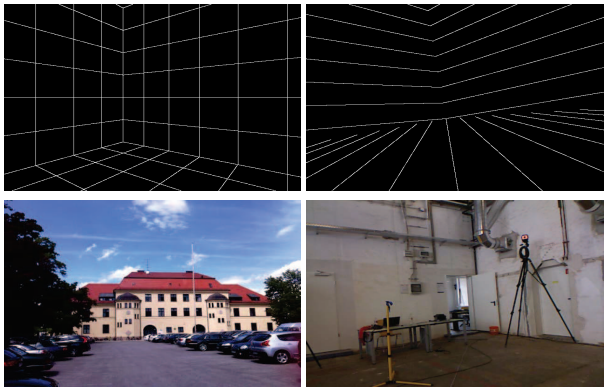


Figure 5. Synthetic and real datasets. Two synthetic datasets were generated for the experiments: images of the Manhattan world scene (top-left) and images of the non-Manhattan world scene composed of three mutually non-orthogonal VPs (top-right). The real datasets are the Metaio (bottom-left) and TUM (bottom-right) datasets. Many spurious parallel line segments are observed from the cars, trees, and clouds.

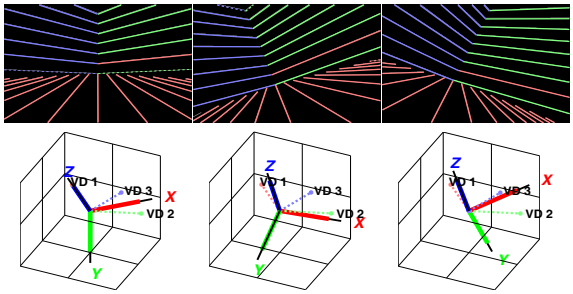
The RANSAC-based VP detector presented in Section 4.2 is used to detect new VPs. We run the detector every β frame or in the case where the number of the VPs is less than two. In addition, spurious VPs should be deleted in the state vector since they degrade the accuracy and robustness of the estimation. We observed that they tend to rotate by some degrees from their initial orientation or have very few line measurements. Thus, we delete any VP that has very few measurements in the current frame or rotates more than γ degrees from the initial direction. In the experiments, we set β and γ to 30 and 10, respectively.

5. Results

To evaluate the proposed joint estimation method, we used several synthetic and real datasets. The images in the synthetic datasets were generated at a resolution of 640×480 and composed of two different kinds of scene: the Manhattan world scene and the non-Manhattan world scene. For the evaluation in real-world tasks, we used the Metaio [13] and TUM [21] datasets as shown in Fig. 5. The Metaio dataset was captured at a resolution of 480×360 and the TUM dataset at 640×480. In this section, we compare the proposed method to the state-of-the-art methods, Exhaust[4] and BnB[5], and a method that detects VPs from the VP extraction method of [22] without the Manhattan world assumption and then matches the nearest VP, called Nearest.

Table 1. Performance comparison of the synthetic datasets. The performance is evaluated by the average ratio of rotation error for each rotation of 10, 50, 100, and 150 degrees (%).

Algorithm	Manhattan			Non-Manhattan
	$\sigma = 0.5$	$\sigma = 1.0$	$\sigma = 2.0$	$\sigma = 0.5$
Nearest	14.21	29.23	44.42	14.87
Exhaust[4]	2.80	2.94	3.18	-
BnB[5]	3.25	3.78	3.64	-
Proposed	**0.76**	**0.94**	**1.19**	**0.70**

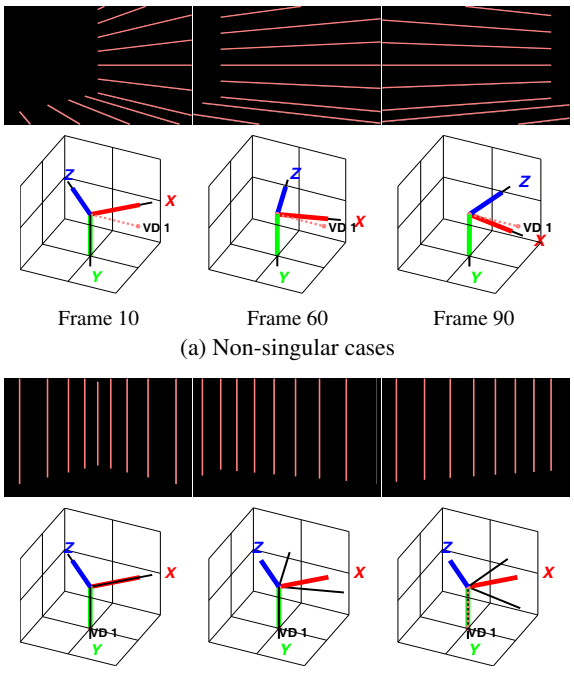


Frame 10 Frame 45 Frame 120

Figure 6. Results of the proposed method in the non-Manhattan world scene. The line measurements (colored lines) extracted in the input images (top) are used to jointly estimate the camera orientation and VPs (bottom). In the bottom images, the red, green, and blue solid lines represent the orientation of the camera, the colored dotted lines are VD vectors, and the black solid lines represent the ground truth pose of the camera orientation. The blue line represents the head of the camera. Each of the colored lines in the top images corresponds to the VD of the same color.

5.1. Synthetic datasets

For fair comparison, we generated image sequences of a synthesized scene containing three mutually orthogonal VPs because some existing methods assume the Manhattan world. Gaussian noise was added to the end-point positions of line measurements. Accordingly, we ran the experiment 100 times and evaluated the average errors of the rotation estimates. The error was defined by the ratio of the rotation error for each rotation of 10, 50, 100, and 150 degrees as in a manner similar to the method introduced in [10]. Table 1 shows the performance comparison for the synthetic Manhattan world scene with different levels of noise. The results show that the Nearest was very susceptible to noisy measurements whereas the Exhaust and the BnB increased the accuracy considerably since the Manhattan world constraint was enforced. On the contrary, the proposed method did not enforce the Manhattan world constraint but yielded the more accurate results than the other existing methods. Since only noisy line segments were given as measurements without any spurious lines in this experiment, these results experimentally verify that the proposed method, which is designed to estimate the camera orientation and VPs jointly and consider the uncertainties of camera motion and measurements, is accurate and robust to noisy measurements.



Frame 10 Frame 60 Frame 90

(a) Non-singular cases

Frame 10 Frame 60 Frame 90

(b) Singular cases

Figure 7. Singularity of the proposed method. The non-singular VD is not the same as the axis of the camera rotation in (a). In these cases, the camera orientation is estimated well although only one VP is used for the estimation. On the other hand, the VD estimated in (b) and the axis of the camera rotation look in the same direction. The estimated orientation (the red, green, blue solid lines in the bottom) is fixed over all the frames although the actual orientation (black solid lines in the bottom) is changing.

On the other hand, to demonstrate the applicability of the proposed method in a non-Manhattan world scene, we synthesized an image sequence of the non-Manhattan world scene containing three mutually non-orthogonal VPs. The camera was rotated with yawing, pitching, and rolling. We also added Gaussian noise ($\sigma = 0.5$ pixels) to the end-points of the line measurements. The experimental results are presented in Fig. 6 and the camera orientation estimate is compared with the ground truth in Table 1. The results demonstrate that the proposed method still remains an accurate and robust performance although the Manhattan world constraint is not enforced.

However, when the proposed method estimates camera orientation using only one VP[2], we can encounter a problem, called *singularity*. If the axis of the camera rotation is equal to the VD, the estimation is not aware of any change in the rotation. Here, the direction denotes *a singular direction*. This property can be explained by using Eq. (5). Regardless of the amount of rotation, a VD is always orthogonal to the normal of the great circle corresponding to

[2]Note that any methods cannot work with only one VP but the proposed method can work with only one VP except the singularity case.

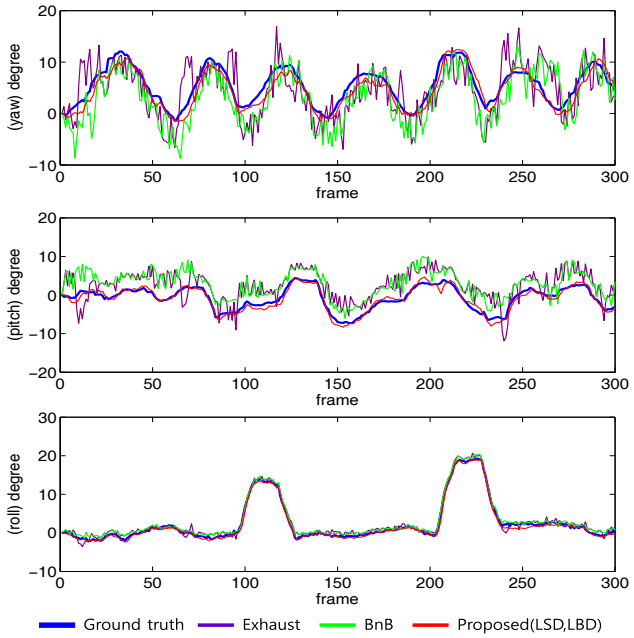


Figure 8. Camera orientation estimates of the several selected methods, the proposed method and the existing methods [4, 5], for the sequence, Metaio-100. The rotation estimate of the proposed method is more accurate and robust than the others.

the VD if the VD is the singular direction. Since all innovation errors from the measurements become zero according to the orthogonality, the camera orientation is not updated but fixed despite the fact that the camera rotates. Figure 7 shows the non-singular and singular cases, illustrating that only one VD is sufficient to successfully estimate the state if the VD is different from the singular direction, while the estimation fails when a VD is equal to the singular direction . Therefore, we need at least two different VPs in order to successfully complete the estimation regardless of the singularity. Nevertheless, it should be noted that the proposed method is still superior to other methods in the sense that our method performs using one or more non-orthogonal VPs, while the other methods for improving accuracy [4, 5] require three orthogonal VPs. In addition, since the feature management technique detects and adds new VPs to the state, the singularity can be handled well in practical applications of the proposed method.

5.2. Real datasets

We used the Metaio and TUM dataset for evaluating the performance of our method in real-world image sequences. The image sequences comprised complex scene structures and hence provided many spurious lines as well as noisy measurements. The existing methods based on the Manhattan world assumption [4, 5] suffer from those factors because the three orthogonal sets of parallel lines are not dominantly seen in the image and thus the factors do not allow the correct triplet of VPs to be found. For this reason, their rotation estimates fluctuate as shown in Fig. 8. In contrast, the proposed method gives a reasonably accurate result in practice for the following reasons. First, the joint estimation method can estimate the camera orientation accurately although noisy measurements are given, as explained in Section 5.1. In addition, the proposed feature management and the RANSAC-based outlier rejection techniques efficiently handle spurious lines and incorrect line correspondences. Unlike the existing methods that find the rotation with the maximum number of inliers and therefore include several spurious lines, the proposed techniques are intended to maintain only valid VPs and line features and exclude the spurious lines and VPs. These techniques contribute greatly to achieving a robust and accurate performance in real-world image sequences.

Table 2 shows the performance evaluation for each real dataset. We ran the methods over 300 frames of each dataset in common. We manually initialized and fixed orthogonal VPs for running the Exhaust and the BnB, while the Nearest and the proposed method automatically detected VPs regardless of the orthogonality constraint. In the experiments, we considered the EDLines and LSD detector as the line segment detectors, and the simplified and the LBD descriptor-based methods as the line segment trackers. As in Table 2, the proposed methods provided a better performance than the other methods. In particular, the method using the LSD detector and the LBD descriptor-based line tracking achieved significant accuracy improvements because more accurate measurements could be obtained by the line detector and more reliable line features can be tracked for a long time by the descriptor. On the other hand, the method using the EDLines detector and the simplified line tracking was less accurate since the EDLines detector provided more noisy measurements. However, the method still produced the better performance in some sequences, Metaio-100 and 101, and was computationally efficient. Therefore, the method is suitable for the real-time performance.

The BnB-based method [5] yielded the best performance in the image sequences, the Metaio-102 and 103, where most of the lines extracted from the line detector were successfully clustered into an orthogonal triplet of VPs. However, the estimation of this method diverged in the image sequence, the TUM-Hemisphere since many spurious lines and measurement noise affect the estimation severely. In contrast, the performance of the proposed method is accurate consistently for all the sequences. This means that the proposed method is superior to the other methods in term of accuracy and robustness in practical applications.

5.3. Computation time

The proposed method was tested on an Intel i7 3.4 GHz CPU and Matlab using a single core. The line detectors

Table 2. Performance comparison for the Metaio and TUM datasets. Red denotes the best results and blue denotes the second best.

Algorithm	Metaio-100	Metaio-101	Metaio-102	Metaio-103	TUM-Hemi.	TUM-Pio.
Nearest (LSD)	49.49	41.56	38.43	43.51	68.39	54.60
Exhaust[4] (LSD)	20.03	17.97	8.51	11.39	18.66	17.53
BnB[5] (LSD)	14.21	13.44	**6.91**	**6.62**	41.61	17.16
Proposed (EDLines,Simplified)	13.97	12.17	12.43	8.81	21.06	18.40
Proposed (EDLines,LBD)	10.49	12.09	9.05	7.56	19.36	17.42
Proposed (LSD,Simplified)	**9.42**	**9.46**	8.31	7.40	**18.08**	**15.17**
Proposed (LSD,LBD)	**8.17**	**9.19**	**7.47**	**7.32**	**17.30**	**11.94**

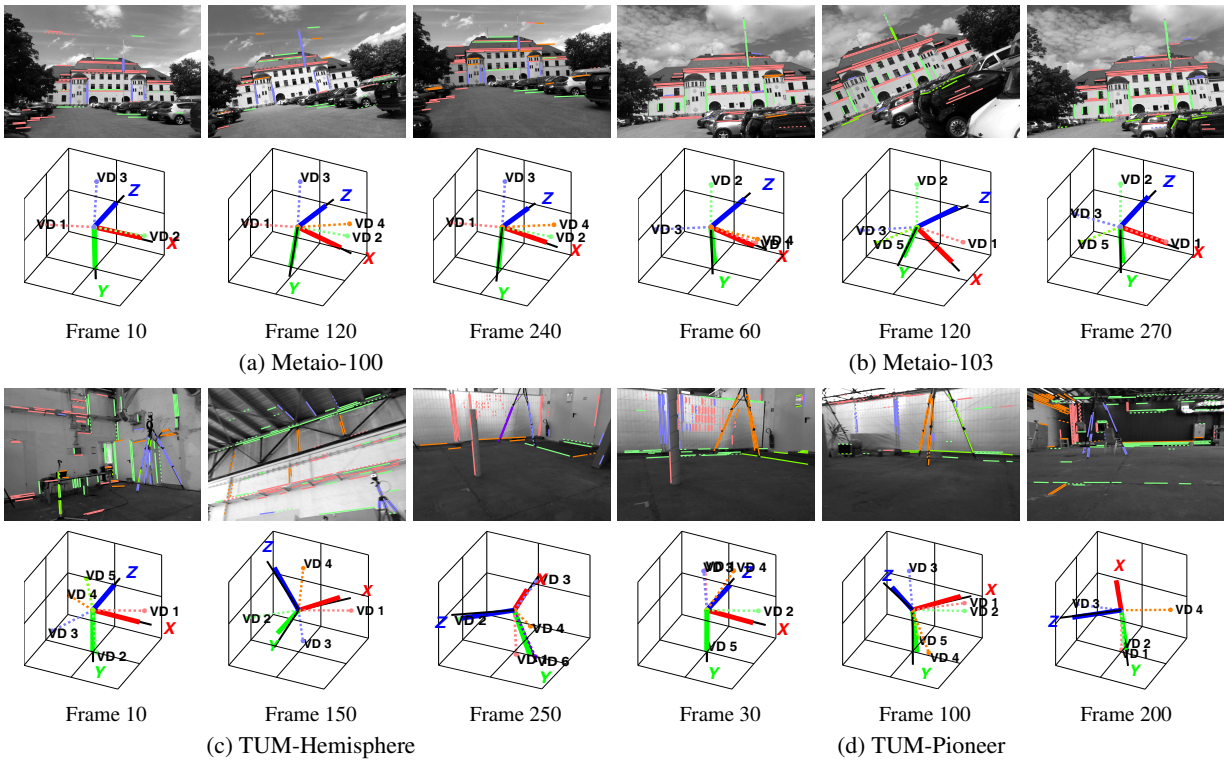


Figure 9. The colored lines on the input images (first row) are the measurements corresponding to the estimated VPs (dotted lines in the second row). In all the datasets, the proposed method well estimates the camera orientation (red, green, and blue solid lines in the second row) as compared with the ground truth pose (three black solid lines).

[2, 23] were run on C as compiled with the mex command. When employing the EDLines detector and the simplified line segment tracking, the proposed method can run in real time; it takes about 40 ms for an average of 76 line features at each frame. Most of the processing time is spent on searching measurements. The searching process has a complexity of $O(nm)$ since it matches n line features with the nearest m lines. Practically, the actual number of m was much smaller than n in the experiments. Thus, the processing time is almost linear in the number of the line features.

6. Conclusion

The estimation of camera orientation through VPs is a difficult problem because of noisy and spurious line segments, high computational complexity, and the geometric constraint. In contrast to the previous methods that are dependent on finding a triplet of orthogonal VPs under the Manhattan world assumption, in the proposed method camera orientation and VPs are jointly estimated based on nonlinear Bayesian filtering. The proposed method enhances the robustness to measurement noise and overcomes the limited scene constraint by the virtue of the joint estimation technique. Therefore, our method can be robustly used in many practical applications in general environments. The proposed joint estimation method actually showed an outstanding performance as compared to the state-of-the-art methods as performing highly accurate estimation even if noisy measurements and spurious line segments were obtained. The code of our method is available from [1].

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2012R1A1A1010871), Global Frontier Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Plannig (NRF-2013M3A6A3075453), and ICT R&D program of MSIP/IITP [B0101-15-0552, Development of Predictive Visual Intelligence Technology].

References

[1] http://cvl.gist.ac.kr/project/.

[2] C. Akinlar and C. Topal. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011.

[3] M. E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *CVPR*, 2000.

[4] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. *IJRR*, 31(1):63–81, 2012.

[5] J.-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *CVPR*, 2012.

[6] R. Cipolla, T. Drummond, and D. P. Robertson. Camera calibration from vanishing points in image of architectural scenes. In *BMVC*, 1999.

[7] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel. 1-point ransac for ekf-based structure from motion. In *IROS*, 2009.

[8] W. Elloumi, S. Treuillet, and R. Leconge. Real-time camera orientation estimation based on vanishing point tracking under manhattan world assumption. *Journal of Real-Time Image Processing*, pages 1–16, 2014.

[9] B. Fan, F. Wu, and Z. Hu. Robust line matching through line–point invariants. *Pattern Recognition*, 45(2):794–805, 2012.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[11] J.-E. Gomez-Balderas, P. Castillo, J. A. Guerrero, and R. Lozano. Vision based tracking for a quadrotor using vanishing points. *Journal of Intelligent and Robotic Systems*, 65(1-4):361–371, 2012.

[12] P. Holoborodko. Noise robust gradient operators. http://www.holoborodko.com/pavel/image-processing/edge-detection/, 2009.

[13] D. Kurz, P. G. Meier, A. Plopski, and G. Klinker. An outdoor ground truth evaluation dataset for sensor-aided visual handheld camera localization. In *ISMAR*, 2013.

[14] J. Lezama, R. Grompone von Gioi, G. Randall, and J.-M. Morel. Finding vanishing points via point alignments in image primal and dual domains. In *CVPR*, 2014.

[15] R. Pflugfelder and H. Bischof. Online auto-calibration in man-made world. *Digital Image Computing: Techniques and Applications (DICTA)*, page 75, 2005.

[16] E. Rondon, L.-R. Garcia-Carrillo, and I. Fantoni. Vision-based altitude, position and speed regulation of a quadrotor rotorcraft. In *IROS*, 2010.

[17] C. Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10):647–655, 2002.

[18] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, Hoboken (N.J.), 2006.

[19] S. N. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In *Trends and Topics in Computer Vision*, pages 267–281. Springer, 2012.

[20] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. *ACM Transactions on Graphics*, 27(5), 2008.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012.

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

[23] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *TPAMI*, 32(4):722–732, 2010.

[24] Z. Wang, F. Wu, and Z. Hu. Msld: A robust descriptor for line matching. *Pattern Recognition*, 42(5):941–953, 2009.

[25] Y. Xu, C. Park, and S. Oh. A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In *CVPR*, 2013.

[26] L. Zhang and R. Koch. An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794–805, 2013.