

1-Point Ransac Based Robust Visual Odometry

Nguyen Van Cuong¹, Moon Beom Heo², Gyu-In Jee^{1†}

¹Department of Electronics Engineering, Konkuk University, Seoul 143-701, Korea

²Satellite Navigation Team, Korea Aerospace Research Institute, Daejeon 305-806, Korea

ABSTRACT

Many of the current visual odometry algorithms suffer from some extreme limitations such as requiring a high amount of computation time, complex algorithms, and not working in urban environments. In this paper, we present an approach that can solve all the above problems using a single camera. Using a planar motion assumption and Ackermann's principle of motion, we construct the vehicle's motion model as a circular planar motion (2DOF). Then, we adopt a 1-point method to improve the Ransac algorithm and the relative motion estimation. In the Ransac algorithm, we use a 1-point method to generate the hypothesis and then adopt the Levenberg-Marquardt method to minimize the geometric error function and verify inliers. In motion estimation, we combine the 1-point method with a simple least-square minimization solution to handle cases in which only a few feature points are present. The 1-point method is the key to speed up our visual odometry application to real-time systems. Finally, a Bundle Adjustment algorithm is adopted to refine the pose estimation. The results on real datasets in urban dynamic environments demonstrate the effectiveness of our proposed algorithm.

Keywords: Ackermann's principle; 1-point method; rotation estimation; Bundle Adjustment

1. INTRODUCTION

Visual odometry (VO) is the process of estimating the motion of a camera using the image sequence. The image sequence can be generated from a single vision (single or omnidirectional camera), stereo vision or multi-camera system. Many researchers around the world have been trying to develop the best motion estimation system possible. Some criteria such as low cost, computation time, simplicity of the algorithm, and whether it works in dynamic environments can be used to evaluate whether a visual odometry system is well-developed or not. Many algorithms were successfully produced to be applied in real-time urban environments (Parra et al. 2008, Tardif et al. 2008, Scaramuzza et al. 2008). Using stereo vision systems or multi-camera systems, we can directly estimate the rotation and translation of the camera. However, the computation time and expense drawbacks of using more than one camera. Many algorithms using these systems can

still only work off-line or at a low frame-rate because of the computation time. So our work mainly focuses on a single camera.

In stereo vision, the scale ambiguity will be removed, and both rotation and translation can be directly computed from stereo frames (Howard 2008). An iterated sigma point Kalman filter was employed to combine with a Ransac-based outlier rejection scheme to robustly estimate vehicle motion in dynamic environments (Kitt et al. 2010). The stereo features were separated into two groups based on their usefulness (Kaess et al. 2009), with one group used to recover rotation using two-point Ransac, and another used to recover translation using one-point Ransac. An upgrade (Golban et al. 2012) significantly improves the existing VO algorithms by using the rank transform to change the image illumination, and proposing a new consistency check to reliably remove outliers.

In single vision based visual odometry, the omnidirectional Ladybug camera was used to produce a VO application in urban environments (Tardif et al. 2008), with distance up to 2.5 kilometers. He adopted a 5-point preemptive Ransac to estimate motion. Scaramuzza & Fraundorfer (2009) also used an omnidirectional camera system and proposed the lowest Ransac model, called 1-point Ransac, to estimate

Received Mar 9, 2011 Revised Apr 26, 2013 Accepted Apr 29, 2013

[†]Corresponding Author

E-mail: gjee@konkuk.ac.kr

Tel: +82-02-450-3070 Fax: +82-2-3437-5235

motion. Scale ambiguity in single vision is also strictly solved (Kitt et al. 2011, Scaramuzza et al. 2009).

Ransac (Fischler et al. 1981) has been established as the standard method for motion estimation in the presence of outliers. Ransac works by generating model hypotheses from randomly minimal sample datasets and verifying them on the whole datasets. The hypothesis that shows the highest consensus with other data is selected as the correct solution. The limitation of Ransac is that it requires exponentially more iterations depending on the number of points required to generate a hypothesis. As such, using so many iterations will slow down the Ransac algorithm. For this reason, there is a high interest in using a minimal parameterization of the Ransac model. As we know, six degrees of freedom (DOF) motion can be estimated from a minimum of five correspondences. Several 5-point minimal solvers were proposed (Triggs 2000, Nister 2003). Later, some attempts were conducted to reduce the number of necessary motion parameters. A new minimal Ransac method was proposed (Naroditsky et al. 2012), the three-plus-one method, to compute the relative pose in monocular visual odometry by using three image correspondences and a common direction. Using this four-point Ransac, they also give a successful 6DOF VO. A two-point Ransac was presented to recover rotation and a one-point Ransac to recover translation (Kaess et al. 2009). In the case of planar motion assumption, a one-point minimal Ransac solver was proposed (Scaramuzza et al 2009). Performance evaluations of five-, two-, and one-point Ransac algorithms were also shown by Scaramuzza (2011). As another approach to 1-point Ransac, a novel combination of Ransac plus Extended Kalman Filter was presented (Civera et al. 2010), which uses the available prior probabilistic information from the EKF in the Ransac hypothesis stage.

In this paper, we concentrate on solving the extremely difficult challenges in visual odometry such as high amount of computation time, complexity of the algorithm and the fact that it doesn't work in urban environments. Our contribution is that we employ a 1-point method to improve the Ransac algorithm and the relative motion estimation. We combine this 1-point method with the iterative estimation method to generate a lowest Ransac model, which is also called 1-point Ransac. This Ransac contributes two important gains: the first is removing all outliers on moving objects, which helps our algorithm to be applied in urban environments without considering the effect of moving objects in motion estimation. The second is using the smallest number of necessary iterations corresponding to 1-point but still guaranteeing that a correct solution can

be computed, which helps our algorithm to remarkably reduce computation time. In addition, in the motion estimation algorithm, by combining the 1-point method with a simple least-squares linear solution, the complexity of our algorithm is reduced. Furthermore, our algorithm can deal with situations in which just a few feature points are present, unlike many algorithms, which can fail due an insufficient number of points. The paper will proceed as follows: In Part II, we focus on describing every step of our proposed VO algorithm in detail. Finally, in Parts III and IV, we present our experimental results and conclusions.

2. PROPOSED VISUAL ODOMETRY ALGORITHM

In this section, we detail each step of our proposed visual odometry algorithm as shown in Fig. 2.

2.1 Feature detection and matching

Currently, SIFT (Lowe 2004) and SURF (Bay et al. 2006) are widely used in visual odometry. We tested both SIFT and SURF, and found that the latter performed better. Our experiments proved that, with the same data and parameters, the trajectories resulting from SURF could fit the ground truth better than the trajectories resulted from SIFT. This is most likely because of SURF detector can

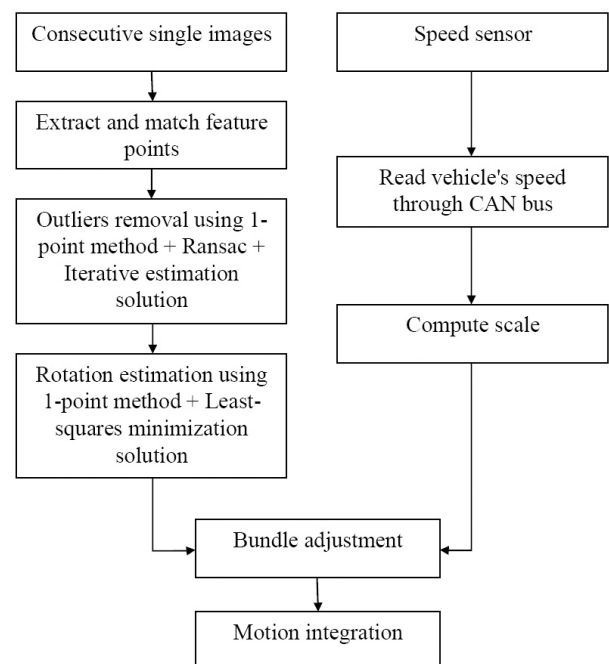


Fig. 2. Flowchart of our proposed visual odometry algorithm.

extract more reliable, robust and invariant feature points. We utilized a traditional approach for SURF matching, which was computing the squared Euclidean distance between SURF descriptor vectors. However, we made an improvement to increase the matching accuracy and to reduce the searching time. We constrain our searching space for finding correspondences. For every feature point on the previous image, we define a square region on the current image that is within a relatively high distance of 50 pixels to the epipolar line. To find its correspondence in the current image, instead of searching all points on the whole current image as in the original matching algorithms, we just searched in the defined region. In this paper, the SURF descriptor vectors with the smallest Euclidean distance are temporarily considered as the corresponding features. Although the descriptors-based feature matching is reliable, false correspondences are still unavoidable. So we added more constraints in order to improve the matching precision.

- Uniqueness constraint: one feature can only match with another one.
- Threshold of Euclidean distance: a correspondence can be accepted if its vector distance or score is less than a threshold, which was set to 1 in all our experiments.
- Distance constraint: for two different correspondences $p_1 \leftrightarrow p'_1$ and $p_2 \leftrightarrow p'_2$, the distance between p_1 and p_2 should be almost the same to as distance between p'_1 and p'_2 .

2.2 Outlier removal algorithm

After performing the first step, we get a set of correspondences between 2 consecutive images. But this set also includes some false correspondences. The possible reasons for outliers are illumination, image blur, change in view or image noise. Such outliers can cause significant error in motion estimation. Currently, the most common method used to remove outliers is Ransac. Most of the current Ransac algorithms have the shortcoming of significantly increasing the computational cost, because so many iterations are used that it slows down the VO algorithm. For this reason, it is extremely important to find the minimal Ransac model. The number of iterations needed to guarantee a correct solution is measured as follows (Bay et al. 2006):

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \quad (1)$$

where, p is the probability of success, ε is the percentage of outliers, and s is a randomly selected number of

Table 1. Number of necessary iterations.

Number of points	Number of iterations
1	7
2	16
3	34
4	71
5	145
6	292
7	587
8	1177

points. By assuming that $\varepsilon=50\%$ and $p = 99\%$, we arrive at Table 1, which shows the number of necessary iterations corresponding to the chosen set of correspondences.

As shown in Table 1, the minimal number of necessary iterations is 7, which corresponds to using only 1 point. This is the lowest model parameterization possible and results in the Ransac algorithm that is the most efficient, and that reduces the computational time to the shortest time possible. For this reason, we adopt the 1-point method and Ransac, also known as 1-point Ransac (Scaramuzza et al. 2009), using the constraints of circular planar motion. Most of the streets in urban environments are approximately planar, and thus we can consider the motion of a camera as planar motion. Furthermore, according to Ackermann's steering principle (Siegwart & Nourbakhsh 2004), a point, known as Instantaneous Center of Rotation (ICR), must exist around each wheel so that the vehicle can follow a circular motion. Thus, we can describe the motion of a camera as circular motion.

Generally, more than 5 correspondences are needed to estimate 6-DOF motion between two images. But if we assume that the motion of the camera is similar to planar motion, the two relative poses can be represented by three unknowns (3DOF). As shown in Fig. 1 to find the relative motion between two consecutive positions O_{k+1} and O_{k+1} , we need to estimate 3 unknown parameters $[\Phi, \theta, \rho]$, where Φ is moving direction, ρ is scale and θ is rotation on the 2D plane. Moreover, using the benefit of circular motion, we add one additional constraint, which is $\theta = 2\Phi$. Finally, we just need to estimate 2 unknowns: rotation angle θ and scale ρ . We will describe how to get scale later. In the 1-point Ransac algorithm, we temporarily set scale at 1, and thus only θ needs to be estimated.

In planar motion, the vehicle always keeps a constant distance from the road when moving. We consider that the camera frame axes are parallel to the ground plane. So, based on the above constraints, the rotation and translation of the camera are represented as follows:

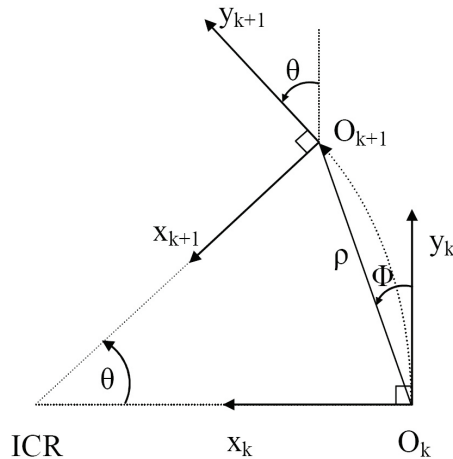


Fig. 1. Motion relation between two consecutive positions under planar motion assumption and Ackermann's principle.

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \rho \begin{bmatrix} \sin\frac{\theta}{2} \\ \cos\frac{\theta}{2} \\ 0 \end{bmatrix} \quad (2)$$

From a 3D point in space, we set p' and p as its corresponding image coordinates in the previous image and current image, respectively. Their normalized coordinates $P'=[x', y', z']^T$ and $P=[x, y, z]^T$ will be computed as follows:

$$P' = K^{-1}p' \text{ and } P = K^{-1}p \quad (3)$$

where, K is the calibration matrix.

Using the epipolar constraint, the defining equation for the essential matrix is:

$$P'^T E P = 0 \quad (4)$$

where, E is defined as $[T]_x R$ and $[T]_x$ is the skew symmetric matrix of T .

Using Eqs. (2), (3) and (4), we obtain the homogeneous equation as follows:

$$(y'z + z'y)\sin\frac{\theta}{2} + (z'x - x'z)\cos\frac{\theta}{2} = 0 \quad (5)$$

Finally, we can compute the rotation angle from Eq. (5) as:

$$\theta = 2 * \tan^{-1}\left(\frac{x'z - z'x}{y'z + z'y}\right) \quad (6)$$

From Eq. (6), we can see that the rotation between

two relative positions can be estimated by using only one correspondence. We call this method the 1-point method, and this method will become very useful in cases where only several feature points, or even one feature point, are present. The equations above are valid only when the position of the camera must satisfy Ackermann's principle that requires the camera to be mounted along the back-wheel axis of the car and the front-side axis of the camera to be perpendicular to it. In practice, however, cars moving in on-road environments have a small steering angle, which results in a big radius of curvature. Thus we can place the camera anywhere on the car and satisfy the requirement that its front-side axis be perpendicular to the back-wheel axis.

Continuously, by applying the 1-point method above, we can parameterize the lowest Ransac model, which is known as 1-point Ransac. Firstly, in each Ransac iteration, by selecting a correspondence randomly, we compute the rotation angle θ between two relative positions using Eq. (6). Secondly, we calculate the fundamental matrix (referred to as a hypothesis):

$$F = K'^{-T} E K^{-1} = K'^{-T} [T]_x R K^{-1} = K'^{-T} \begin{bmatrix} 0 & 0 & \cos\frac{\theta}{2} \\ 0 & 0 & \sin\frac{\theta}{2} \\ -\cos\frac{\theta}{2} & \sin\frac{\theta}{2} & 0 \end{bmatrix} K^{-1} \quad (7)$$

Finally, we check all remaining correspondences and verify them as inliers if they satisfy the computed hypothesis. To do this, for each remaining correspondence, we employ the Levenberg-Marquardt method to minimize the following geometric error cost function given the calculated hypothesis and a correspondence:

$$f(\hat{p}', \hat{p}) = \underset{(\hat{p}', \hat{p})}{\operatorname{argmin}} \|p' - \hat{p}'\|^2 + \|p - \hat{p}\|^2 \quad (8)$$

subject to $\hat{p}'^T F \hat{p} = 0$. Where, p' , p are the given corresponding feature points on camera images. \hat{p}' , \hat{p} are the projected points on images, respectively.

We verify a correspondence as an inlier if the reprojection errors $\|p' - \hat{p}'\|$ and $\|p - \hat{p}\|$ are less than 1. For feature points on moving objects, their reprojection errors will be very large, so most of those points will be verified as outliers and are removed. Therefore our algorithm can be applied in urban environments without considering the effect of moving objects.

2.3 Rotation estimation

This step is a process to estimate the rotation angle between two consecutive positions. As shown in Eq. (6), we can acquire the rotation angle θ by using only one correspondence. However, the question is how to estimate the rotation angle if there is more than one correspondence. To solve this problem, we combine the 1-point method with a least-squares solution of homogeneous equation. From Eq. (5), we arrive at the following homogeneous equation:

$$DX = \begin{bmatrix} (y'_1 z_1 + z'_1 y_1) & (z'_1 x_1 - x'_1 z_1) \\ \vdots & \vdots \\ (y'_n z_n + z'_n y_n) & (z'_n x_n - x'_n z_n) \end{bmatrix} X = 0 \quad (9)$$

where, $X = \begin{bmatrix} \sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix}$ is an unknown variable and n is the number of correspondences.

The problem in Eq. (9) is corresponding to find X that minimizes $\|DX\|$, subject to $\|X\|=1$. So we employ a simple least-squares minimization solution, based on Singular Value Decomposition (SVD) algorithm, to minimize the cost function as follows:

$$f(\hat{X}) = \operatorname{argmin}_{\hat{X}} \sum_{i=0}^n (D_i X)^2 \quad (10)$$

where, D_i is the i -th row of A . Finally, θ will be obtained from:

$$\hat{X} = \begin{bmatrix} \sin(\frac{\theta_{est}}{2}) \\ \cos(\frac{\theta_{est}}{2}) \end{bmatrix} \quad (11)$$

Above, we show how to combine the 1-point method with a least-squares solution to estimate the rotation between two relative images based on a simple linear solution. If there is only one correspondence, Eq. (9) will be become Eq. (6) and we will only use a 1-point method to find the rotation angle. If there is more than one correspondence, Eq. (10) will be used to find the most optimal rotation angle. Generally, the higher the number of correspondences, the more accurate the rotation estimation is.

2.4 Scale measure

Single vision is simpler and less expensive than stereo vision, but it unavoidably suffers from scale ambiguity. This is the most extreme limitation of monocular-based motion

estimation. A simple approach to solving scale ambiguity is using additional sensors such as GPS or IMU. In our system, to reduce the complexity of the algorithm and decrease the computation time, we use an additional speed sensor to read the speed of a vehicle through CAN bus. Thus the vehicle scale will be calculated as $\rho = v\Delta t$, where v is the vehicle speed and Δt is time difference.

2.5 Bundle Adjustment

This step is a process to refine the egomotion estimation. Bundle Adjustment is a non-linear minimization problem providing a maximum likelihood estimate. In our VO application, we employ a Bundle Adjustment algorithm to minimize reprojection error given a set of corresponding pairs to optimize the rotation and translation.

The problem we want to solve can be formulated as follows:

$$\operatorname{argmin}_{(R,T)} \sum_{i=0}^n (\|x_i^{pre} - f(R^{pre}, T^{pre}, X_i)\|^2 + \|x_i^{cur} - f(R^{cur}, T^{cur}, X_i)\|^2) \quad (12)$$

where, $f(R^{pre}, T^{pre}, X_i)$ is the transformation function that projects a 3D point X_i into pixel x_i^L on the previous image. A set of matched point pairs (x_i^{pre}, x_i^{cur}) is given and is filtered out from the 1-point Ransac.

To minimize the non-linear least-squares problem in Eq.(12), we used the Levenberg-Marquardt algorithm. This algorithm requires an initial guess for R and T . It can achieve a fast convergence to the minimum estimate based on a good initial guess. As an initial guess, we used the linear solution in Eq. (10) to provide the initial rotation, and used scale from vehicle sensors to provide the initial translation.

2.6 Motion integration

Motion integration is a process of computing the absolute egomotion of the current position to the initial position using the rotation θ and the scale ρ estimated above. We define z -axis as directed to the front side of camera, which x -axis is horizontally directed to the right side of camera. The odometry pose $[x, z, \theta]^T$ at the k -th position in planar motion (3DOF) will be computed as follows:

$$\begin{cases} x_k = x_{k-1} + \rho \sin(\theta_{k-1} + \theta/2) \\ z_k = z_{k-1} + \rho \cos(\theta_{k-1} + \theta/2) \\ \theta_k = \theta_{k-1} + \theta \end{cases} \quad (13)$$

Table2. Computation time of Ransac algorithms and error of estimated rotation with respect to true rotation (1 degree).

	1-point Ransac	2-point Ransac
Computation time (s)	0.4656	1.502
Rotation error (deg)	0.043	0.175

Table3. Computation time of Ransac algorithms and error of estimated rotation with respect to true rotation (3 degrees).

	1-point Ransac	2-point Ransac
Computation time (s)	0.2947	0.8827
Rotation error (deg)	0.191	0.025

Table4. Computation time of Ransac algorithms and error of estimated rotation with respect to true rotation (5 degrees).

	1-point Ransac	2-point Ransac
Computation time (s)	0.1706	0.5397
Rotation error (deg)	0.794	0.196

3. EXPERIMENTS

3.1 Comparison with 2-point Ransac

In this section, we evaluate the performance of our 1-point Ransac with the 2-point Ransac. The evaluation was done by comparing computation time and estimation accuracy. Two images were acquired at two different positions with scale of 1 meter. We tested the accuracy of rotation estimation on {1,3,5}-degree rotation. This range of scale and rotation θ in this simulation was obtained based on what we experienced with the real data from our experiments. At 10Hz image capture rate we saw that the biggest rotation is less than 5 degrees. According to the settings above, the relative pose between two positions is:

$$[\rho, \theta] = [1.0 \text{ m}, \{1,3,5\} \text{ deg}] \quad (14)$$

To obtain statistically meaningful results, we executed two Ransac algorithms 500 times.

Computation time and error of estimated rotation θ were represented in Tables 2, 3, 4. As a result, it was found that computation time for the 1-point Ransac was greatly smaller than computation time for the 2-point Ransac. This is true because, as analyzed in Part II, the number of required iterations of 1-point Ransac is smaller than those of 2-point Ransac. In terms of rotation estimation, the estimated rotation using 2-point Ransac was better than the estimated rotation using 1-point Ransac. At this point, the reader may be wondering why 2-point Ransac performs better than 1-point Ransac. The reason is that 1-point

Ransac assumes the motion of the camera as a circular motion, which is just the approximately real motion of the camera according to Ackermann's principle. Also, 1-point Ransac just uses one correspondence while 2-point Ransac uses two correspondences to estimate motion. Nevertheless we can realize that errors of estimated rotation using 1-point Ransac were not big and we can accept these errors. Particularly in cases where the vehicle is straightly moving or slightly turning, errors of estimated rotation using 1-point Ransac were very small, as shown in Tables 2 and 3. In summary, using 1-point Ransac can reduce computation time significantly compared to other Ransac algorithms. Furthermore, its smallest model parameterization allows us to deal with situations in which only one correspondence is present.

3.2 Experiments on real image sequences

Our experiments were performed on a real car equipped with a Bumblebee2 camera, a speed sensor, and a GPS-RTK/INS system. The Bumblebee camera and speed sensor were used to provide image sequence and vehicle speed, respectively. We just used left-side images whose resolution was 640x480 at 10Hz capture rate. The car speed ranged between from 0 to 65 km/h. The GPS device was used to provide the reference trajectory with a highly precise accuracy, up to sub-meter. The dataset was taken in real traffic in Seoul city. As such, many moving objects such as cars, bus, pedestrians were present. We tested on two different sequences. The results are shown in Figs. 3, 4. In both figures, the top shows the GPS reference trajectory on Google Earth, the middle shows the comparison between our estimated VO trajectory (red) and the reference trajectory (green) given by the GPS-RTK/INS system, and the bottom shows the number of inliers detected by our 1-point Ransac. Observing on two experiments, the estimated trajectory was well fitted with the reference trajectory. When the car is straightly moving or slightly turning, we have many inliers, up to several hundreds. In special cases when the car is significantly turning, there are just several inliers occurring, even only one inlier, as shown in Fig. 5. However, our VO algorithm can recover an accurate result in those cases. To evaluate the accuracy of visual odometry, we measured RMS error and drift error at the ending position over the total distance as shown in Table 5. As a result, we found that the estimated VO trajectories were similar to the trajectories given by GPS-RTK/INS system and the drift errors are small, and these results demonstrate the reliable effectiveness of our proposed VO algorithm.

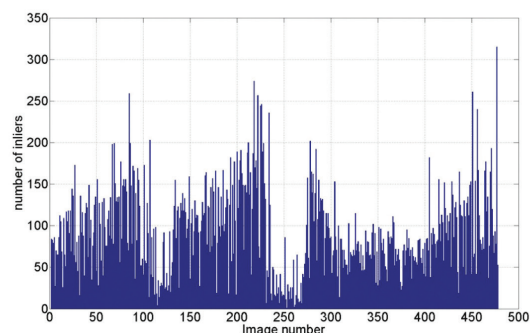
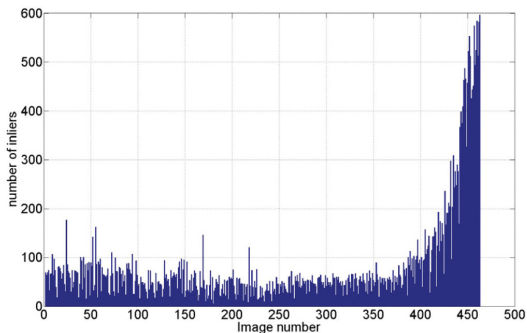
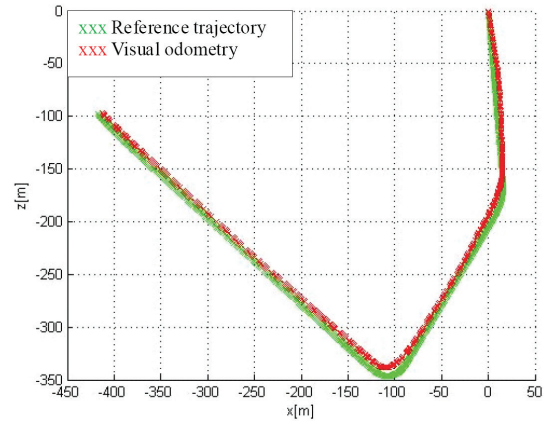
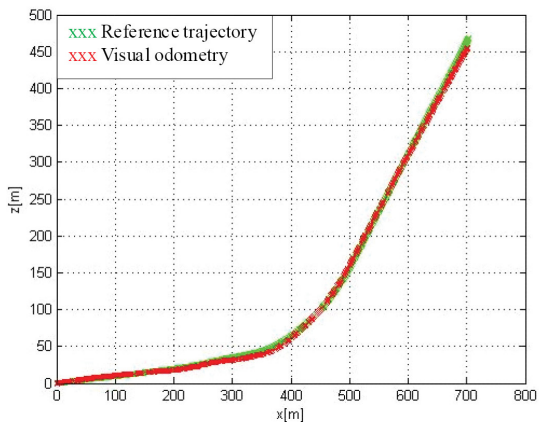


Fig. 3. Experiment on the 1st sequence. Top shows trajectory in the city on a Google Map. Middle shows the result of our VO trajectory (red), compared to the reference trajectory (green) given by the GPS-RTK/INS system. Bottom shows the number of detected inliers.

Fig. 4. Experiment on the 2nd sequence. Top shows trajectory in the city on a Google Map. Middle shows the result of our VO trajectory (red), compared to the reference trajectory (green) given by the GPS-RTK/INS system. Bottom shows the number of detected inliers.

4. CONCLUSIONS

In this paper, we propose a robust and efficient visual odometry algorithm that can be reliably applied in urban environments. Using the planar motion assumption and Ackermann’s steering principle, we adopted a 1-point method to improve the Ransac algorithm and the relative motion estimation. 1-point Ransac allows us to significantly reduce computation time and provides an accurate estimated rotation with small error. Also, in

Table5. This table shows drift error and RMS error in position for two sequences.

Sequence	Distance (m)	Drift error (%)	RMS (m)	images
1	940.0706	1.43	9.12	463
2	789.0895	0.63	5.75	478

motion estimation, we employ a 1-point method to provide a simple linear solution that helps reduce the complexity of our VO algorithm. Our motion estimation algorithm can deal cases in which only a few points are present, unlike

many algorithms, which can fail when there are not enough points. Our VO algorithm was simple because we just computed the motion based on consecutive images. We did not use the previous poses or structure to refine the current pose. These benefits will accelerate our visual odometry for further applications to real-time systems.

In future work, we will focus to solving the scale ambiguity without using other sensors, which is still a challenge in single vision based visual odometry. In addition, we are going to use a Ladybug 3 camera instead of a Bumblebee2 to improve the accuracy.

ACKNOWLEDGMENTS

This research was supported by a grant from the Transportation System Innovation Program (TSIP) funded by Ministry of Land, Transport and Maritime Affairs (MLTM) under the Korean government.

REFERENCES

- Bay, H., Tuytelaars, T., & Van Gool, L. 2006, "SURF: Speeded Up Robust Features", in *Computer Vision - ECCV 2006*, 3951, 404-417.
- Civera, J., Grasa, O. G., Davison, A. J., & Montiel, J. M. M. 2010, 1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry, *Journal of Field Robotics*, 27, 609-631.
- Fischler, M. & Bolles, R. C. 1981, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of ACM* 26, 24, 381-395.
- Golban, C., Szakats, I., & Nedeveschi, S. 2012, Stereo based visual odometry in difficult traffic scenes, In *Proceedings of Intelligent Vehicles Symposium*, 736-741.
- Howard, A. 2008, Real-time stereo visual odometry for autonomous ground vehicles, In *Proceedings of the international conference on intelligent robots and systems*, 3946-3952.
- Kaess, M., Ni, K., & Dellaert, F. 2009, Flow separation for fast and robust stereo odometry, in *Proceedings of the international conference on Robotics and Automation*, 3539-3544.
- Kitt, B., Rehder, J., & Chambers, A. 2011, Monocular visual odometry using a planar road model to solve scale ambiguity, In *Proceedings of the 5th European Conference on Mobile Robots*.
- Kitt, B., Geiger, A., & Lategahn, H. 2010, Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme, in *Proceedings of Intelligent Vehicles Symposium*, 486-492.
- Lowe, D. G. 2004, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60, 91-110.
- Naroditsky, O., Zhou, S., Roumeliotis, I., & Daniilidis, K. 2012, Two Efficient Solutions for Visual Odometry Using Directional Correspondence, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 818-824.
- Nister, D. 2003, An efficient solution to the five-point relative pose problem, In *Proceedings of Computer Vision and Pattern Recognition*, 2, II-195-202.
- Parra, I., Sotelo, M., & Vlacic, L. 2008, Robust visual odometry for complex urban environments, In *Proceedings of Intelligent Vehicles Symposium*, 440-445.
- Scaramuzza, D. 2011, Performance evaluation of 1-point RANSAC visual odometry, *J. Field Robot*, 28, 792-811.
- Scaramuzza, D., Fraundorfer, F., Pollefeys, M., & Siegwart, R. 2009, Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints, In *Proceedings of the 12th international conference on Computer Vision*, 1413-1419.
- Scaramuzza, D., Fraundorfer, F., & Siegwart, R. 2009, Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC, *IEEE international conference on Robotics and Automation*, Kobe, Japan, 4293-4299.
- Scaramuzza, D. & Siegwart, R. 2008, Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles, *IEEE Transactions on Robotics*, 24, 1015-1026.
- Siegwart, R. & Nourbakhsh, R. 2004, *Introduction to Autonomous Mobile Robots*, MIT Press.
- Tardif, J., Pavlidis, Y., & Daniilidis, K. 2008, Monocular visual odometry in urban environments using an omnidirectional camera, In *Proceedings of the international conference on intelligent robots and systems*, 2531-2538.
- Triggs, B. 2000, Routines for relative pose of two calibrated cameras from 5 points, Technical report, INRIA.



Nguyen Van Cuong has received B.E. in Computer Science and Engineering from Ho Chi Minh University of Technology, Viet Nam 2010. He is currently pursuing Master in Electronics Engineering in Konkuk University, Korea. His research interests include Computer Vision, SLAM, Embedded System.



Moon Beom Heo has received Ph.D. degree in the Mechanical and Aerospace Engineering from the Illinois Institute of Technology, Chicago in 2004. He is currently a head of the Satellite Navigation Department at the Korea Aerospace Research Institute, Korea. His work is focused on Global Navigation Satellite Systems and their augmentation.



Gyu-In Jee has received his Ph.D. degree in Systems Engineering from Case Western Reserve University in 1989. He is currently a professor in the Department of Electronics Engineering at Konkuk University, Seoul, Korea. His research interests include GPS positioning, GPS/INS system, GPS anti-jamming, Vision/Lidar, SLAM, Autonomous Ground Vehicle.