Visual Odometry with Reduced-Iterative Optimization Method for Automobile

Chuho Yi, Jungwon Cho

Abstract Background/Objectives: ***Visual odometry (vehicle motion estimation) is a fundamental technology for applications based on vision. Most applications require the latency to be very short.***

Methods/Statistical analysis: ***That is, it is necessary to reduce the computational load or the number of data points included in any calculation. However, this makes algorithms sensitive to noise, so most researchers have focused on methods to reduce their computational complexity instead.***

Findings: ***In this paper, we propose an improved visual odometry method for automotive applications. Generally, the optical flows of visual odometry algorithms have many outliers when applied to the automotive field; therefore, most approaches select N arbitrary points and then perform Random Sample Consensus. The error distance is calculated over the entire dataset, and inliers are selected based on a threshold. However, this increases the computational complexity, and the number of iterations required of this type of iterative execution over N points increases with N.***

Improvements/Applications: ***In this paper, we decrease this computational cost by improving execution time. Our method is a modification of the efficient perspective-N-point method, which uses four representative control points.***

Keywords: ***Visual odometry, RANSAC, EPnP (Efficient Perspective-N-Point), Reduced-iterative optimization, Automobile.***

I. INTRODUCTION

In camera-based applications for vehicles, the visual odometry algorithm is very important for estimating the motion of a vehicle. Visual odometry involves measuring how far the camera on the vehicle has moved based on the optical flow of N matched points [1,2]. Matthies proposed a method for measuring the movement of a robot on Mars, and this is used in the Mars Exploration Rovers [4]. Since then, several applications of visual odometry have been proposed [2,3]. In general, after randomly selecting N arbitrary points, the perspective-N-point (PnP) method is used to estimate the movement of the camera by iterating an optimization calculation [12,13].

In this paper, we provide a new visual odometry method for automotive applications. In the case of vehicles, the amount of movement in the forward direction (z) of the camera is large, and odometry estimates are further confused by various environmental factors, such as moving cars, trees shaking in the wind, and so on. Geiger addresses this difficulty by selecting N random points in the first step, then estimating the motion of the camera. The error distance over the complete dataset is estimated based on the result of the first step, and outliers are identified as errors greater than a certain threshold. Finally, the motion of the camera is estimated using all data excluding these outliers [5].

However, the fundamental issue to Geiger's method is that the computational time increases with increases in the number of points used at the time of optimization. Lepetit introduced a method to reduce computation by calculating four control points. Each point represents a subset of all data points, and control point calculation is optimized [6], where N points are expressed in a 12×12 matrix. This method, called the efficient PnP (EPnP) method, greatly reduces the calculation load.

Visual odometry is used to estimate how far a camera has moved between a previous position and a current position. The distance is estimated based on connected optical flow points, and the method approximates surrounding static obstacles and the greater environment using a triangulation method. Because using all flow points would make the computation intractable, a subsample is selected to estimate motion using the least-squares method for the calculation.

The method of selecting inliers and outliers has a decisive influence on the accuracy of visual odometry. However, in an automobile platform, this is difficult to do mainly because of variation in the natural environment, such as moving trees, other vehicles, pedestrians, changes in light, and so forth.

Initially, Geiger used a relatively low threshold when applying the PnP algorithm to select inliers. Then the accuracy of the estimation was improved by conducting a second-order calculation with the selected inliers and applying the PnP algorithm with a higher threshold [7]. Although good results can be obtained applying this method in automotive platforms, it has a relatively large computational load because iteration increases by two stages RANSAC. In this paper, we propose a method to reduce the computational complexity of this two-stage method. Generally, when selecting N points and inversing the induced matrix, the computation load increases with the number of points and the number of iterations in the PnP algorithm.

To address this problem, we first apply the EPnP method [6]. We calculate the values of the four control points that we use to represent a random sample of points,

Revised Manuscript Received on May 22, 2019.

Chuho Yi, Department of Electronics, Dong Seoul University, Seongnam, South Korea

Jungwon Cho, Department of Computer Education, Jeju National University, Jeju, South Korea (Corresponding author)

E-mail:jwcho@jejunu.ac.kr

then use the least-squares method to calculate the estimated motion. In this way, because the dimensions of the matrix are fixed at 12×12, the computational complexity of matrix inversion can be greatly reduced. Secondly, we apply Ferraz's method [8], which uses the residual values calculated while inverting the matrix to resolve rapidly increasing computational cost. That is, we apply this method to identify inliers and outliers. By using residual values, it is possible to eliminate the process of re-projecting all points, thus significantly reducing computational complexity.

In this paper, we use an outlier rejection scheme to remove outliers. We repeat it to identify second-order inliers, which are then used to obtain the estimated values. The computational load is slightly larger in this case than when outlier rejection is applied only once. However, compared to the conventional two-stage PnP method, the computational complexity of the estimation process is reduced to about 23%.

II. PROPOSED METHOD FOR VISUAL ODOMETRY

Our proposed method is described below.

Algorithm I

1. First, four representative control points are calculated with N points. To conduct visual odometry, it is necessary to identify points that correspond between a previous image and a current, shifted, image. According to current methods, such points should be in static areas. This will improve the accuracy of outlier detection, namely, the identification of points that have been moved or have been detected incorrectly, during the estimation stage. The points observed in the world coordinate system are expressed as $X^w = \{X_1^w, X_2^w, \dots, X_n^w\}$.Then these points are projected to $X^c = \{X_1^c, X_2^c, \dots, X_n^c\}$ in the image coordinate system by applying the camera internal calibration matrix *A*. These N points are calculated by combining the four representative control points used to increase the efficiency of computation. The calculation is

$$X_i^w = AX_i^c = A\sum_{j=1}^{4} \alpha_{ij} c_j^c \qquad (1)$$

where α_{ij} is the homogeneous barycentric coordinate and $\sum_{j=1}^{4} \alpha_{ij} = 1$. In this way, visual odometry can be estimated by solving an estimation problem based on four control points with a particular representation.

2. The optimization proceeds based on the four control points obtained in Step 1 above. Then camera motion is estimated in one step. By writing a more specific version of equation (1), we obtain the following rearrangement:

$$\begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} \qquad (2)$$

Where f_u and f_v are the horizontal and vertical focal lengths of the lens and u_c and v_c are the coordinates of the optical center of the camera's sensor.These parameters are expressed in terms of$A$. The term on the right side of the equation expresses$c_j^c = [x_j^c \quad y_j^c \quad z_j^c]^T$.Combining this and equation (2), we can summarize the expression as follows:

$$\sum_{j=1}^{4} \alpha_{ij} f_u x_j^c + \alpha_{ij}(u_c - u_i) z_j^c = 0 \qquad (3)$$

$$\sum_{j=1}^{4} \alpha_{ij} f_v y_j^c + \alpha_{ij}(v_c - v_i) z_j^c = 0 \qquad (4)$$

whereu_i and $v_i$are the currently observed positions of the points on the image. When expressions (3) and (4) are concatenated and represented by the matrix for all N points, we obtain the following:

$$\Omega Z = 0 \qquad (5)$$

where $\Omega$is a 2N×12 matrix and Z is the vector of the control points, $Z = [c_1^c c_2^c c_3^c c_4^c]^T$, a 12×1 vector.Finally, the visual odometry estimation is reduced to the problem of finding the null space of the matrix Ω.To estimate the value of camera motion, we obtain the least-squares estimation from $\Omega^T\Omega$, which becomes a 12×12 matrix. Thus the calculation cost becomes very small, on the order of O(N).We estimate the motion by applying Gauss-Newton Optimization to perform an efficient searchto estimate motion[9].

3. We calculate the error distance for all points, based on the results of the motion estimation obtained during the first stage. Then we perform SVD on $\Omega^T\Omega$calculated in Step 2, to obtain its eigenvalues. Next, we use a bucketing method to verify that the selected points are uniformly distributed over the entire image. The number of points selected, n, is decided by the designer in advance.

4. The error points of Ω are used to identify the outliers based on the obtained error distance. After calculating the error ε in Step 3, we decide whether the selected points are to be considered inliers or outliers when estimating camera motion. The following two methods are used to identify outliers. Initially, we calculate the reference error values for all points. Then we rearrange the calculated error values in descending order and classify a certain percentage as outliers. $E = \{\varepsilon_1, \varepsilon_1, \dots, \varepsilon_n\}$is the set of error values of point ε, and the equation for assigning a certain proportion of points as outliers is

$$D(\varepsilon_i) = \begin{cases} 1, & \text{sort}(\varepsilon_i)/n \geq r_{\text{inliers}} \\ 0, & \text{sort}(\varepsilon_i)/n < r_{\text{inliers}} \end{cases} \qquad (6)$$

where the$D(\cdot)$function determines whether a value is an inlier or outlier, and r_{inliers} is the threshold value that determines which points are inliers in set E, which is the sorted error set. We used a threshold of 0.5 in this paper.

Next, if the maximum error between the inliers is larger than a threshold valueτ_1, which is set by the designer, optimization is executed again, this time on the inliers. This procedure will not be repeated more than three to five times. Finally, the number of outliers identified is n_o.

5. Four new control points are calculated again using the $(n - n_o)$inliers identified in Step 4 to estimate the camera motion more accurately.

6. The optimization progresses based on the four control points obtained in the previous step and the visual odometry result calculated in Step 2. Camera motion is estimated in the same way, and we optimized the newly calculated control points using the method described in Step 2. The threshold to be used in the second estimation is set to be much smaller than the previous threshold value τ_1. (thus, $\tau_2 \ll \tau_1$)

III. EXPERIMENTS

We validated the proposed method using the KITTI open dataset [10,11]. We installed the dataset, which was collect using a stereo car camera. In Figure 1, the data acquired by the left camera is a continuous image. Applying our method, we can estimate the distance traveled by the camera in a relatively small number of iterations compared to the existing RANSAC method. Hence, the computational load is relatively small compared to the method based on selecting approximately nine points and then applying RANSAC in two stages. Our method has an average calculation time of about 22% (less than 1/5) that of the previous two-stage RANSAC method.

This open dataset has been developed to enable researchers to evaluate automobile algorithms, including those requiring input from a camera, laser, or radar sensor. The dataset provides an accurate ground truth because the authors installed a Velodyne laser scanner and GPS localization system in addition to a stereo camera, which was mounted on a sedan vehicle. The dataset was captured in rural areas and on the highways of Karlsruhe, a mid-sized city. A large number of vehicles and pedestrians are captured in the image data. We used the ground truth data provided in the dataset to evaluate the algorithm.

We used 11 stereo camera sequences from the dataset. These sequences can be used to evaluate translational and rotational errors. Figure 1 shows example images from the dataset.

Figure 1. Various examples of KITTI open dataset.

These data were collected in various environments, such as in the city and along highways, as shown in Figure 1. The dataset also includes images taken under these circumstances at low speed, high speed, moving forward, and with high-varying rotation. Hence, the dataset contains appropriate sequences for evaluating the proposed algorithm. The images used in this paper were captured using a stereo camera and consist of 1241×376 pixels, excluding the parts of the image that were not necessary when applying the algorithm.

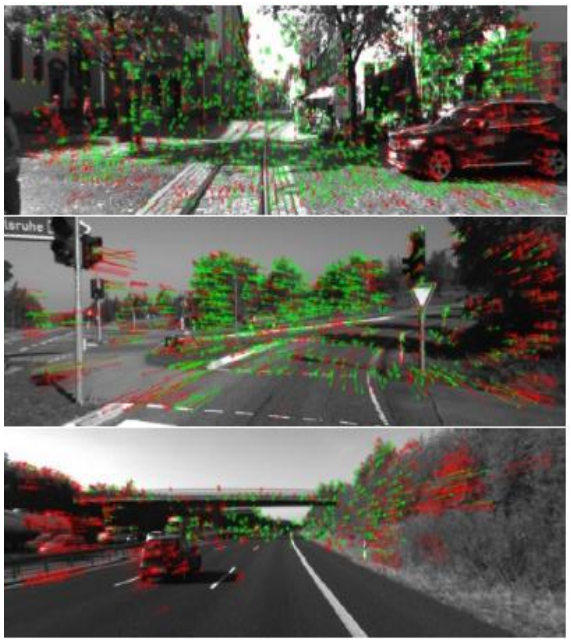

Figure 2. Results of inlier (green) and outlier (red) determination.

Figure 2 shows the results of the outlier determination procedure proposed in this paper. The first image shows the results from a low-speed situation in the city. The middle image is taken when the vehicle is turning in an urban environment. The last image shows the outliers identified when the vehicle is on a highway. In this figure, we can confirm that the results corresponding to moving cars and pedestrians, false tracking points, and distorted points near the camera are classified as outliers. It is possible to estimate how far the vehicle has moved between the previous and current images by applying our visual odometry method because the motion of the camera is compensated for by the inliers. In other words, if the effects of the camera movement are removed, there will still be a difference between a moving object and a stationary object according to our algorithm. Figure 3 shows the results obtained after correcting for motion of the camera.

Figure 3. Results of compensating for optical flow using estimated camera motion.

The images in Figure 3 were calculated based on an

image taken at the same time as the images in Figure 2. In the image at top, we can confirm that the pedestrian seen at the left side of the image and the car at the right are moving. In the middle image, although the vehicle is turning, the compensation renders almost all of the points visible. We can confirm how far the car has moved at high speed using the final image.

Table 1. Results comparing our method to previous method

Sequence Number	Previous Method(Two-stage RANSAC)			Proposed Method		
	Rotational Error(deg/m)	Translational Error(%)	Computation Time(msec)	Rotational Error(deg/m)	Translational Error(%)	Computation Time(msec)
1	2.34	0.0137	23.84	2.47	0.0139	6.21
2	2.62	0.0119	26.07	2.73	0.0124	5.73
3	2.49	0.0124	24.21	2.51	0.0142	5.39
4	2.51	0.0162	24.93	2.62	0.0173	4.86
5	2.27	0.0117	25.34	2.46	0.0137	5.17
6	2.73	0.0125	24.18	2.82	0.0135	5.84
7	2.48	0.0134	25.76	2.61	0.0146	6.18
8	2.31	0.0148	23.45	2.48	0.0152	4.98
9	2.55	0.0112	24.67	2.64	0.0138	5.36
10	2.42	0.0124	25.19	2.53	0.0136	5.84
11	2.64	0.0137	23.74	2.72	0.0144	5.42
Average	**2.49**	**0.0131**	**24.67**	**2.60**	**0.0142**	**5.54**

Table 1 compares the results obtained using the previous two-stage RANSAC methodversus our method. There is a performance degradation of about 4.41% (0.0131 to 0.0142 deg/m) in the case of the rotational error and 8.39% (2.49 to 2.60%) for translational error. These noise components are compensated for to some extent during the estimation process, when calculating the control points while including some outliers. However, the computational time decreases by about 22.45% (24.67 to 5.54 msec). Hence, our proposed method is very efficient, because the calculation is about 4.45 times faster than the previous method.

IV. CONCLUSION

We propose a method to eliminate the iteration component of a standard visual odometry algorithm for automotive applications, and thus reduce the computational load required by introducing a two-step estimation procedure. The reason for dividing the computation into two stages is to exclude the large number of outliers that generally occur in automotive applications. Then we improve the efficiency of the computation by calculating four control points to represent all of the data points. This method enables us to obtain similar motion estimation results while drastically improving the calculation time compared to conventional methods.

REFERENCES

1. M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. of the ACM. 24 (6), pp. 381-395, 1981.
2. D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I - The First 30 Years and Fundamentals," IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011.
3. F. Fraundorfer and D. Scaramuzza, "Visual Odometry: Part II - Matching, Robustness, and Applications," IEEE Robotics and Automation Magazine, Volume 19, issue 2, 2012.
4. M. Maimone, Y.Cheng, and L. Matthies, "Two years of Visual Odometry on the Mars Exploration Rovers," Journal of Field Robotics. 24 (3), pp. 169~186, 2007.
5. Andreas Geiger, Julius Ziegler, and Christoph Stiller, "StereoScan: Dense 3d Reconstruction in Real-time," IEEE Intelligent Vehicles Symposium(IV), 2011.
6. V. Lepetit, F. Moreno-Noguer, and Pascal Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," International Journal of Computer Vision, 2008.
7. B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," In Intelligent Vehicles Symposium (IV), pp. 486-492, 2010.
8. L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Very fast solution to the PnP problem with algebraic outlier rejection," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 501-508, 2014.
9. Å. Björck, Numerical methods for least squares problems, Society for Industrial and Applied Mathematics, 1996.
10. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
11. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," International Journal of Robotics Research (IJRR), 2013.
12. X. S.Gao, X. R.Hou, J.Tang, and H. F.Cheng, "Complete solution classification for the perspective-three-point problem," IEEE transactions on pattern analysis and machine intelligence, 25(8), 930-943, 2003.
13. A.Penate-Sanchez, J.Andrade-Cetto, and F.Moreno-Noguer, "Exhaustive linearization for robust camera pose and focal length estimation," IEEE transactions on pattern analysis and machine intelligence, 35(10), 2387-2400, 2013.