Abstract: Recently, visual odometry has gained more attention due to its application to autonomous vehicles and robots. In this paper, we propose visual odometry using a single camera. This configuration is economically more attractive than other popular configurations using stereo cameras. To estimate frame-to-frame motion more accurately, we thoroughly utilize prior knowledge on our operating space and motion. Our visual odometry assumes that its operating space is locally flat so that its movement undergoes planar motion. From this assumption, we can adopt the novel normalized 1-point algorithm to estimate relative pose between a pair of images. Moreover, under this assumption, we can resolve scale ambiguity, the most serious problem in monocular visual odometry. To estimate scale of motion, we introduce the other 1-point algorithm derived from planar homography of the ground plane. Two 1-point algorithms are the key to accelerate our visual odometry for its further application to real-time systems. Our experiments on synthetic data verified that the normalized 1-point algorithm was more accurate than other known method. Moreover, our experiments on real data presented effectiveness of our proposed visual odometry.

Keywords: Visual Odometry, Monocular, Planar Motion, Relative Pose Estimation, Scale Estimation, 1-point RANSAC

1. INTRODUCTION

Vision-based localization has been investigated in robotics and computer vision from their beginning due to its attractiveness. Recently, its popularity and importance have increased because cameras became more advanced and inexpensive. Among vision-based localization, approaches with natural landmarks have more advantages than the others with artificial landmarks. Vision-based localization with natural landmarks does not need to install distinct landmarks which require more cost and visually degrade its operating space. However, such approaches with natural landmarks should adopt more advanced and complex algorithms to extract and recognize effective landmarks from natural scenes. Therefore, it is necessary to investigate more on vision-based localization with natural landmarks.

Vision-based localization with natural landmarks can be categorized by single-query and multiple-query approaches. Table 1 contains significant works based on this classification. To estimate pose of a camera, the single-query approaches utilize only the current image (a query) incorporated with a previously generated image database [1], [2]. In contrast, the multiple-query approaches use not only the current query but also its previous queries instead of the prepared database [3], [4], [5]. Therefore, the multiple-query approaches are usually more robust in varying environments which inevitably contain illumination change, moving objects, and so on. An outdoor environment is a representative example for such varying situation. Visual odometry is a simple but effective solution for the outdoor since it is usually large and navigation in it rarely has a path with loops. Recently, visual odometry has gained more attention because of its further application to autonomous vehicles and robots in the outdoor.

1.1 Previous Works on Visual Odometry

Visual odometry has been tackled as a simpler version of visual SLAM (Simultaneous Localization and Mapping) or SfM (Structure from Motion) problem. In other words, visual odometry has been regarded as visual SLAM without loop closure (especially in robotics) or SfM without bundle adjustment (especially in computer vision). Therefore, the most important step in visual odometry is a process to estimate camera motion between adjacent image pairs, whose general formulation is to estimate 6 DoF relative pose between them. There have been two representative techniques to solve this problem. The first techniques are based on epipolar (sometimes trifocal or quadfocal) geometry [3], [6], [7], [8], [9]. With stereo cameras, the problem becomes the PnP (Perspective N-point) problem, which is solved by 3-point algorithm. With monocular cameras, camera motion is extracted from an essential matrix or planar homography, which are estimated by 5-point and 4-point algorithms, respectively. The second techniques utilize tracking which usually consists of prediction and correction [4], [5], [8], [10]. Well-known examples are Bayesian filters such as Kalman filter with proper motion and observation models [4]. Our visual odometry follows the first approach based on epipolar geometry because this approach easily accepts large numbers of feature points enough to achieve better accuracy [11]. Especially, our method is based on 1-point algorithm proposed by Scaramuzza et al. [7], which is briefly described in Section
2. Scaramuzza’s 1-point algorithm assumes that a camera undergoes 2D planar and circular motion (2 DoF), not 3D general motion (6 DoF). Planar motion is generally valid in indoor and on-road environments even though such environments are not completely flat, but only locally flat. This assumption makes the relative pose estimation much simpler and provides more accurate results with less computation time. We performed preliminary analysis on it [12].

1.2 Previous Works on Scale Estimation

Monocular cameras are simple and inexpensive compared with stereo cameras, but monocular motion estimation inevitably suffers from scale ambiguity. Scale ambiguity is the most critical limitation in monocular motion estimation, thus many researches have been executed to resolve it. Davison et al. [4] used a predefined landmark at beginning of their visual SLAM, and Klein and Murray [5] assumed the initial camera movement as a known-length translation. However, these approaches using initial assumptions can face the other problem, scale drift, because their recent scale estimate is strongly based on their previous estimate so that their error can be accumulated over time. Until now, these approaches are available in small-size spaces and motion with loops. Scaramuzza et al. [7] adopted an additional sensor such as a speedometer. However, this approach with extra devices inevitably entails complex systems with more cost. Scaramuzza’s other work using an omni-directional camera [6] utilized planar homography of the ground plane which keeps the constant interval with his camera. His approach does not require initial assumptions or additional devices except only one assumption that the ground is flat. Our approach is also based on planar homography with this assumption so that our camera has the fixed gap with the ground.

1.3 Contributions

Our visual odometry equips with three novel techniques in comparison with Scaramuzza’s visual odometry [7]. First, we introduce the normalized 1-point algorithm to estimate relative pose between image pairs. When we apply more than one correspondence to 1-point algorithm, the original 1-point algorithm generates a least-square solution which minimizes algebraic error. However, algebraic error usually does not mean geometrically meaningful distance. Our normalized 1-point algorithm generates relative pose which minimizes geometric error so that it can give more accurate solution than its original one. Second, our visual odometry utilizes the novel error function to accelerate RANSAC when RANSAC strengthens 1-point algorithm against outliers. This error function does not require complex trigonometric functions in contrast to Scaramuzza’s error function. Lastly, we propose the other 1-point algorithm to estimate scale based on planar homography. Without additional sensors and initial assumptions, our visual odometry can estimate scale using this novel algorithm with RANSAC. Three novel techniques are described in Section 2, respectively. Moreover, their effectiveness was verified by two kinds of experiments, which are presented in Section 3.
2. PROPOSED MONOCULAR VISUAL ODOMETRY

Our monocular visual odometry is based on epipolar geometry using 1-point algorithm under planar motion assumption [7]. It is composed of five steps as shown in Figure 2.

2.1 Feature Extraction and Tracking

The first step is a process to retrieve feature points and find their correspondences. Scaramuzza et al. [7] performed experiments using three different features (Harris corners, KLT, and SIFT) and presented that visual odometry with KLT was the most accurate. We utilize the KLT feature tracker implemented in OpenCV. Our visual odometry extracts Shi and Tomasi’ corners, also known as Good Feature to Track, on the recently acquired image. When it acquires one more image from the camera, it performs Lukas-Kanade optical flow to find local movement of each corner on the newly acquired image. The extracted corners (on the previous image) and their movements (on the current image) consist of feature correspondences between the previous and current images.

2.2 Outlier Rejection

The second step is a process to exclude optical flows which are regardless of camera motion. These wrong optical flows, usually called as outliers, can be generated by poor feature tracking or moving obstacles. Such outliers can cause significant error in estimating camera motion, thus we adopt 1-point algorithm and RANSAC, also known as 1-point RANSAC [7], to reject them.

One-point algorithm calculates relative pose between a pair of images using only one point and its correspondence. Generally, more than 5 points and their correspondences are necessary to estimate motion between two images because relative pose is represented as 6 variables up to scale. However, if a camera undergoes planar motion on flat surfaces, its relative pose is represented as 3 DoF: \([\rho, \phi, \theta]^{\top}\) where \(\rho \) and \(\phi \) is translation (scale and moving direction) in 2D polar coordinate and \(\theta \) is rotation (heading change) on the 2D plane (Refer Figure 1 in [12]). Moreover, if the motion is approximated as circular, an additional constraint, \(\theta = 2\phi \), becomes valid so that its relative pose becomes 2 DoF. Under planar and circular motion assumption, the epipolar constraint, \(X'Ex = 0\), becomes

\[
(xy' - x'y') \cos \phi - (zy' + z'y') \sin \phi = 0, 
\]

where \(x = [x, y, z]^\top \) is an extracted corner on the previous image, and \(x' = [x', y', z']^\top \) is its correspondence on the current image. From the epipolar constraint (1), 1-point algorithm provides relative pose without scale as follows:

\[
\phi = \tan^{-1} \frac{\pm (xy' - x'y)}{\pm (zy' + z'y)}. 
\]

RANSAC (Random Sample Consensus) [13] is a simple but effective algorithm to estimate parameters from data contaminated with outliers. RANSAC is iterations of hypothesis generation and its evaluation. At first, RANSAC generates relative pose (a hypothesis) through 1-point algorithm from a randomly selected correspondence. Next, RANSAC evaluates the estimated pose by counting the number of its supporting correspondences. The supporters are determined as correspondences which have the small amount of error from the estimated pose. Originally, Scaramuzza et al. utilized the error derived from triangular relationship between the estimated pose and correspondence (See Figure 4 in [7]). Their error calculation involves complex operations such as trigonometric functions so that it can degrade computation time of RANSAC. In our visual odometry, we adopt the novel error function as follows:

\[
e(x_i, x'_i; \hat{\phi}) = \frac{s}{\sqrt{s^2 + c^2}} \cos(\hat{\phi}) + \frac{c}{\sqrt{s^2 + c^2}} \sin(\hat{\phi}) \quad (3)
\]

\[
s = x'_i(y_i - x_i) \quad \text{and} \quad c = z'_i(y_i - x_i), \quad (4)
\]

where \(x_i \) is \(i\)-th point, and \(x'_i \) is its correspondence, and \(\hat{\phi} \) is the estimated pose. Our new error function geometrically means

\[
e(x_i, x'_i; \hat{\phi}) = \sin(\phi_i - \hat{\phi}), \quad (5)
\]

where \(\phi_i \) is relative pose which comes from \(i\)-th correspondence. Our error function is possible to computes \(s \) and \(c \) just once during all iterations of RANSAC because they are independent of \(\hat{\phi} \). Moreover, \(\cos(\hat{\phi}) \) and \(\sin(\hat{\phi}) \) can be computed during Equation (2) without trigonometric functions. For these reasons, our error function can accelerate RANSAC much more than Scaramuzza’s error function.

2.3 Relative Pose Estimation

The third step is a process to calculate relative pose from all correspondences except outliers. It is possible to extend 1-point algorithm to \(N\)-point least-square version to get more accurate results. From epipolar constraint (1), a least-square solution is achieved by solving a linear equation as follows:

\[
A a = 0 \quad \text{such that} \quad a = \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}, \quad (6)
\]
where $i$-th row of $A$ by $i$-th correspondence is

$$A_i = [x_1 y_1^i - x_1 y_i, -z_1 y_i - z_1 y_1^i].$$

Our visual odometry adopts normalization to the linear equation (6) in order to improve its accuracy more. If we normalize each row of $A$ to be $|A_i| = 1$, its least-square solution minimizes the geometric error as follows:

$$\hat{\phi} = \arg\min_{\phi} \sum_i \sin^2(\phi - \phi_i),$$

where $\phi_i$ is orientation of motion calculated by $i$-th correspondence. In contrast, the original least-square algorithm finds the solution which minimizes the algebraic error as follows:

$$\hat{a} = \arg\min_{a} \sum_i (A_i a)^2.$$ \hspace{1cm} (9)

Generally, minimizing algebraic error does not guarantee the optimal solution which minimizes geometrically meaningful distance. Figure 4 contains error histograms by the original and normalized 1-point algorithms, respectively. It shows that the normalized 1-point algorithm estimates relative pose more accurately than its original.

Similarly with Scaramuzza et al. [7], our visual odometry performs not only the normalized 1-point algorithm, but also 2-point algorithm to take into account of non-circular motion. In contrast to 1-point algorithm, 2-point algorithm does not ground on circular motion assumption, $\theta = 2\phi$, which can estimate $\phi$ and $\Theta$ through a linear equation [12] or Newton’s iteration method [14]. We also utilize Scaramuzza’s firewall condition to select final relative pose among two results by the normalized 1-point algorithm and 2-point algorithms.

### 2.4 Scale Estimation

The fourth step is a process to calculate scale of relative pose using correspondences on the ground. At first, our visual odometry needs to identify optical flows on the ground. We use Surface Context, proposed by Hoiem et al. [15], which splits the given single image into three geometric regions: ground, sky, and other vertical regions. Since we extract the ground region using Surface Context, optical flows classification is simply done. Figure 3 presents an example of Surface Context on Karlsbushe dataset.

Our visual odometry adopts the novel scale estimation which needs only one correspondence on the ground plane. If the ground is flat and the camera moves on the ground with constant interval, planar homography on the ground plane can describe camera motion without scale ambiguity. Similarly with the epipolar constraint (1), we can derive another constraints from planar homography, $x' = Hx$, as follows [12]:

$$x'y' = x'y\cos \theta + z'y\sin \theta + y'y\frac{d}{d} \sin \phi$$

$$z'y' = -x'y\sin \theta + z'y\cos \theta + y'y\frac{d}{d} \cos \phi,$$ \hspace{1cm} (10)

where $d$ is the fixed height from the ground to the camera. From the constraint with known $\phi$, $\theta$, and $d$, we can lead scale $\rho$ as

$$\rho = \frac{d}{y'y'} \{x'y'\sin(\theta - \phi) + z'y'\cos(\theta - \phi) + x'y'\sin \phi - z'y'\cos \phi\},$$ \hspace{1cm} (12)

which is the other 1-point algorithm which provides scale of relative pose based on planar homography of the ground plane. Similarly with the outlier rejection in Section 2.2.2, RANSAC is utilized to reject inliers which are not consistent with the major ground plane.

### 2.5 Relative Pose Accumulation

The last step is a process to apply the currently estimated relative pose and scale to the previous odometry pose. We follow two conventional coordinate systems to represent relative pose and odometry, respectively. The relative pose is based on the local image coordinate whose $x$-axis is horizontally toward the right side of image, $y$-axis is vertically downward the bottom of image, and $z$-axis is toward the front of image. The odometry pose is represented on the global coordinate whose $x$-axis is the initial heading of camera and $y$-axis is 90 degrees to its right. Rotation in both coordinates follows the right-handed rule. The odometry pose is updated by the relative pose $[\rho, \phi, \theta]^T$ as follows:

$$x'^o_i = x'^o_{i-1} + \rho \cos(\theta'^o_{i-1} - \phi)$$

$$y'^o_i = y'^o_{i-1} + \rho \sin(\theta'^o_{i-1} - \phi)$$

$$\theta'^o_i = \theta'^o_{i-1} - \phi$$ \hspace{1cm} (15)
3. EXPERIMENTS

3.1 Synthetic Data

Our experiments using synthetic data aim to compare accuracy of the normalized 1-point algorithm with other known methods. Five points were randomly generated on the ground plane which is 1.0 meters below the camera. They were observed by the camera whose intrinsic parameters are known as

\[
K = \begin{bmatrix} 100 & 0 & 320 \\ 0 & 100 & 240 \\ 0 & 0 & 1 \end{bmatrix}.
\]

(16)

Two images were acquired at different position and orientation whose relative pose is

\[
[\rho, \phi, \theta]^\top = [1.0m, -10\text{deg}, -20\text{deg}]^\top.
\]

(17)

Each observed point on two images had unbiased Gaussian noise (\(\sigma = 0.5\) pixels). Four algorithms were performed 10³ times to get statistically meaningful results. Figure 4 represents error histograms of estimated \(\theta\) by four algorithms.

As a result, the normalized 1-point algorithm had the least error compared with other three algorithms. Comparison between original and normalized algorithms showed effectiveness of normalization which imposes geometrical meaning on the original algebraic results. Moreover, according to comparison on \(\{5, 2, 1\}\)-point algorithms, we can conclude that more prior knowledge on motion can improve relative pose estimation if the prior knowledge is valid. We could get the similar results on estimated \(\phi\) and also different magnitude of Gaussian noise.

3.2 Real Image Sequences

Our experiments using real data aim to evaluate the proposed visual odometry compared with Scaramuzza’s original version. Two image sequences in Karlsbushe dataset [8] were utilized to quantify accuracy of two algorithms. Karlsbushe dataset was captured by two PointGrey Flea2 cameras whose resolution was 1344x372 with 10Hz frame-rate. We only used left-side images during our experiments and took into account of pitch of the mounted camera, −0.08 radians. We assigned height \(d\) as 1.6 meters, threshold of 1-point RANSAC as 0.3 degrees, and threshold of the firewall condition as 6 degrees. The dataset also provides the ground truth by a highly precise GPS/IMU system, OXTS RT3000. The first image sequence (20100309-0023) had 112 images along with its 93.65-meter trajectory, and the second image sequence (20100309-0019) had 374 images along with its 224.7-meter trajectory. To quantify accuracy of visual odometry, we measured drift error at the end of its trajectory. For statistically meaningful results, we performed visual odometry 20 times on each image sequence and each

<table>
<thead>
<tr>
<th>Original</th>
<th>Estimated Scales</th>
<th>True Scales</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20.21m (21.58%)</td>
<td>1.418m (1.514%)</td>
</tr>
<tr>
<td>Normalized</td>
<td>19.09m (20.38%)</td>
<td>1.588m (1.696%)</td>
</tr>
</tbody>
</table>

Table 1: Drift error on Karlsbushe dataset (20100309-0023): Each drift error is median among 20 trials whose ratio to the overall distance (93.65 meters) is written in parentheses.

<table>
<thead>
<tr>
<th>Original</th>
<th>Estimated Scales</th>
<th>True Scales</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>37.38m (16.56%)</td>
<td>26.99m (11.96%)</td>
</tr>
<tr>
<td>Normalized</td>
<td>28.94m (12.82%)</td>
<td>14.03m (6.216%)</td>
</tr>
</tbody>
</table>

Table 2: Drift error on Karlsbushe dataset (20100309-0019): Each drift error is median among 20 trials whose ratio to the overall distance (225.7 meters) is written in parentheses.

method. Table 1 and 2 present drift error of four configurations, and Figure 5 and 6 show their trajectories with the ground truth.

As a result, visual odometry with the normalized algorithm had less drift error than visual odometry with its original algorithm. Such improvement was more significant (25 to 45%) on the image sequence (20100309-0019) whose trajectory was more complex and longer. However, it is difficult to compare drift error on the other image sequence (20100309-0023) because their difference is only small amount. Inconsistency of two results might result from their complexity of trajectory such as the number of rotation. On the other hand, visual odometry with estimated scales had 10 to 20 meters longer trajectory whose error per frame is almost 0.1 to 0.2 meters.

4. CONCLUSION

In this paper, we propose three novel techniques to improve visual odometry based on 1-point RANSAC [7]. The normalized 1-point algorithm can estimate relative pose more accurately than its original because it min-

![Fig. 5: Trajectories with true scales on Karlsbushe dataset (20100309-0023) whose drift error is median among 20 trajectories](image-url)
Fig. 4: Four histograms (a) – (d) describe error distribution on estimated $\theta$ by four different algorithms.

Fig. 6: Trajectories with true scales on Karlsbushe dataset (20100309-0019) whose drift error is median among 20 trajectories.

imizes geometric error, not algebraic error. Moreover, the fast geometric error function is introduced in 1-point RANSAC to accelerate its computation time. Lastly, we also present the 1-point scale estimation (12) derived from planar homography with planar motion assumption.

As further works, we need to improve our scale estimation because it now had significant error (0.1 to 0.2 meters per frame). Local bundle adjustment or Bayesian filtering can improve our pose estimation and scale estimation more. Besides, it would be a good research to investigate adaptive 1-point RANSAC to deal with varying ratio of outliers and multiple models.

REFERENCES


