Terrain Adaptive Detector Selection for Visual Odometry in Natural Scenes

Kyohei Otsu*a, Masatsugu Otsuki*b, Genya Ishigami*c and Takashi Kubota*b

aThe University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan
bISAS/JAXA, 3-1-1 Yoshinodai, Chuo, Sagamihara, Kanagawa, Japan
cKeio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, Japan
(Received: 8 Apr 2013; Accepted: 22 May 2013)

Pose estimation is one of the important tasks for mobile robots exploring in outdoor environments. Recently, visual odometry has received a lot of attention since its localization is accurate even with low-cost sensors. Furthermore, the technique is not affected by wheel slips, and it can be performed without external infrastructures and preliminary maps. While existing techniques successfully provide good localization of outdoor vehicles, possible failures are not yet fully examined in untextured terrains where feature tracking occasionally fails. This paper proposes an approach to detect interest points from a wide variety of terrains by adaptively selecting algorithms. Experiments show that the approach provides robust and fast interest point detection even in untextured natural scenes.

Keywords: visual odometry; interest point detectors; terrain classification; outdoor environment

1. Introduction

The ability to estimate the motion of a robotic vehicle is important for robot navigation systems. Besides wheel odometry, inertia sensors, GPS, and laser range finders, another informative solution can be provided by using low-cost vision sensors. The technique to estimate motion from images is named visual odometry (VO) in [1]. In recent years, various research groups have made substantial progress in VO for outdoor environments [1–4]. While the developed techniques can localize robots in various environments, outdoor scenery still imposes big challenges on the vision-based algorithms. One significant challenge is the failure of feature tracker. Since most recent algorithms rely on feature tracking in the image sequence, the lack of well-tracked features causes a problem in untextured environments such as sandy terrain [5]. This problem is also complicated because of the parameter tuning problem. Thus, the large diversity of natural terrain causes the failure of an algorithm optimized for a scene if it is used in another scene. Another problem is the computational efficiency which is closely related to the real-time operation. Vision-based algorithms are resource consuming, hence an efficient algorithm is required for processing on an on-board CPU of low performance. These are important problems to be solved for the development of autonomous outdoor vehicles.

The goal of this work is to develop a reliable and efficient VO algorithm for outdoor environments, thereby increasing feasibility of autonomous outdoor navigation. Toward this goal, several challenges should be overcome including the high computational cost and the sensitivity to imaging conditions. As mentioned earlier, the lack of visual features and the large diversity of terrain appearance are big issues to be concerned in outdoor environments. The algorithm

*Corresponding author. Email: kyon@ac.jaxa.jp
should be stable in any environments, robust to any types of terrain, and fast enough to operate in real time. Ultimately, the algorithm may be extended to planetary environments which impose strong limitations. Examples include rugged off-road terrain, no GPS, no prior knowledge, high demand of safety, and strictly limited computational resources.

This paper presents a first step toward this goal. The algorithm proposed in this paper addresses the challenges in an outdoor environment in terms of interest point detection, i.e., finding the good points to be tracked. It is designed to be capable of low-textured terrain with reasonable speed, as well as keeping robustness to large terrain diversity. The contribution of this paper can be summarized as follows:

- This paper proposes a hybrid interest point detector for fast and stable detection in outdoor environments. The detector adopts a rule-based methodology that picks the appropriate detector in accordance with a variety of terrain texture. It is improved by branch prediction using AdaBoost-based classifiers and a saturating counter.
- This paper presents the evaluation of commonly used detectors for the dataset collected at volcanic areas. The dataset comprises more than 1000 stereo image pairs with ground truth.

The rest of this paper is structured as follows: Section 2 discusses the literature on VO and interest point detectors, with focusing on existing VO approaches in natural scenery. Section 3 describes the design of the hybrid detector using branch prediction and machine learning technique. In Section 4, the setup for the comparative evaluation of interest point detectors is detailed and its result are given in Section 5. Finally, Section 6 concludes this paper with the discussion of possible improvements.

2. Related Works

2.1 Visual Odometry

This section describes the overview of recent VO trend, and several attempts to handle crucial untextured terrain in natural scenery.

2.1.1 Overview

The idea of ego-motion estimation from images can be traced back to early 1980’s [6]. Since then, a number of VO algorithms have been developed [1–4, 7–9]. Especially, stereo visual odometry has been successfully used on Mars in the MER mission since early 2004 [2, 5]. However, its use was limited since rover’s low-clock CPU requires up to three minutes to compute ego-motion for each step [5]. Thus, more efficient algorithms are proposed for NASA’s Mars Science Laboratory (MSL) project that was launched in 2012 [8], and for the planned ExoMars rover [9]. Besides the applications in the space robotics, VO has been widely applied to other areas such as ground vehicles [3, 4], aerial robots [10], and underwater vehicles [11]. While most of these implementations are based on feature matching (keeping track of a sparse set of ‘landmarks’), motion estimation can be obtained using optical flow (keeping track of dense brightness patterns). However, the optical flow-based approach is known to be prone to drift error [12].

The feature-based VO can also be divided into two approaches in terms of feature matching procedure. One approach is to find features in one image and track them in the consecutive images using local search techniques (e.g., [7]). The other approach is based on structure-from-motion (SfM) technique, which estimates the relative position of two or more camera frames using feature correspondences among the images (e.g., [1, 3]). Recently, the focus has shifted to the latter approach since it is suitable for large motion estimation and expands VO into large-scale environments [13].
2.1.2 Visual Odometry for Untextured Terrain Appearance

Despite the successful results above, VO occasionally fails in untextured environments which lack sufficient number of interest points. The feature-based VO has the implicit assumption that a terrain exhibits rich texture so that feature points can be easily tracked. However, the MER mission has reported that the rovers on Mars encountered many untextured surfaces where the detector hardly detects visual landmarks on the ground [5].

Several approaches have been performed to address this problem. One approach is to choose a competent detector. Considering the diversity of natural terrain, desirable properties of a detector depends on various conditions such as terrain appearance, image conditions (illumination, blur, etc), and camera’s motion scale. Several scale-invariant detectors, which approximates the Laplacian of Gaussian in an efficient way [14, 15], can solve the scale problem and thereby provide highly accurate matching. The CenSurE detector computes highly efficient approximation of Laplacian with a high degree of localization accuracy [16]. Edge points can also be used for this purpose due to the availability of large amount of points [17]. Another approach is to divide images into square blocks and find the most characteristic point in each region [8, 18]. This approach enables feature detection even in featureless terrains, but it suffers from low matching rate due to using too weak features.

The method proposed in this paper is based on the former approach, i.e., using the appropriate detector. The difference is that the proposed method is a hybrid of multiple detectors which involves several configurations. This method can be robust since it holds various properties suited to various terrains. To avoid increasing computational cost, the proposed method does not run several detectors simultaneously; instead, it adaptively selects a detector according to the current terrain appearance. The selection rule will be detailed in the later section.

2.2 Interest Point Detectors

Interest point detection is the first step of feature-based VO. The detectors are referred as corner detectors (e.g., Moravec [6], Harris [19], Förstner [20], Shi-Tomasi [21], and FAST [22, 23]) and blob detectors (e.g., Laplacian-of-Gaussian (LoG) [24], Difference-of-Gaussian (DoG) [14, 15], Fast Hessian [25], and CenSurE [16]). A corner is defined as the intersection of two or more edges. It shows high repeatability with less computational time but is less distinctive. On the other hand, a blob is an image pattern that is distinctive from its neighborhood in terms of intensity, color, and texture. It has higher distinctiveness but is slower to be detected. Additionally, the blob detectors mentioned above are scale-invariant, which is a preferable property for VO. Each detector has its advantage and disadvantage. A comprehensive study on interest point detectors can be found in [26], and Table 1 summarizes the comparison of detectors popularly used in VO applications [13, 26, 27].

Several research groups have compared the performance of interest point detectors [28–30]. Especially, Gauglitz et al. [30] evaluate the detectors in the context of feature-based visual tracking including VO. From the comprehensive study in terms of image conditions and detector parameters, they found that it is difficult to drive a universally valid detector given the many different conditions of specific applications.

<table>
<thead>
<tr>
<th>Feature Detector</th>
<th>Corner</th>
<th>Blob</th>
<th>Rotation invariant</th>
<th>Scale invariant</th>
<th>Affine invariant</th>
<th>Repeatability</th>
<th>Localization accuracy</th>
<th>Robustness</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris</td>
<td>x</td>
<td>x</td>
<td>+</td>
<td>+++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Shi-Tomasi</td>
<td>x</td>
<td>x</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+++</td>
<td>++</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>FAST</td>
<td>x</td>
<td>x</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>+++</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>DoG</td>
<td>x</td>
<td>x</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Fast Hessian</td>
<td>x</td>
<td>x</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>CenSurE</td>
<td>x</td>
<td>x</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Edge-based</td>
<td>x</td>
<td>x</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
3. Terrain Adaptive Detector Selection

The basic strategy of the proposed method is to select the appropriate interest point detector in accordance with the conditions of current terrain. It is implemented in two steps. At first, it analyzes a camera image and estimates the class of current terrain. The classification is performed by using classifiers based on the AdaBoost framework [31]. According to the classification result, the appropriate detector is selected from the prepared detector set. The selection rule is based on a saturating up-down counter which is used as an efficient implementation of the branch prediction in the computer architecture [32].

3.1 Texture Assessment

Humans can estimate the conditions of terrain only with visual input. This capability makes us possible to perform preliminary action planning. It will be straightforward to think the same functionality aids robots to plan a better strategy beforehand.

In the proposed scheme, the appropriate interest point detector is selected according to the texture of ground surfaces. The terrain class is defined based on the observation of terrain images of Mars-analogue volcanic fields. The description of defined terrain classes is as follows:

- **ROUGH**: This type of terrain contains a large number of interest points. Thus, feature tracking is easily obtained.
- **SMOOTH**: This type of terrain contains a few number of interest points. Thus, feature tracking occasionally fails.

The threshold between two classes is set heuristically based on the availability of 30 successful tracking in an image pair, since such number of tracking can compensate the existence of outliers and bias errors. The ROUGH terrain has large features on the ground surfaces, hence it allows to use a high-speed feature detector in order to gain the total performance efficiency. On the other hand, a detector for the SMOOTH terrain where a few interest points appear on the ground should be sensitive and stable.

The terrain classification of an image is performed in the following procedure: firstly, a captured image is divided into 12 blocks. The uppermost four blocks are removed since the uppermost patches may contain the sky or regions far from the camera. Then, the machine learning-based classifier classifies the rest eight blocks into three classes (ROUGH, SMOOTH, SHADOW), where SHADOW represents the shadow of the vehicle itself (see Figure 1 for the examples). The classifier is trained by the AdaBoost framework [31], which is a general methodology of generating a strong classifier out of a set of weak classifiers. Algorithm 1 shows the pseudocode of the binary AdaBoost classifier. The proposed algorithm enables three-class classification by the combination of three binary classifiers. Let \( i \in C_3 = \{ 1, 2, 3 \} \) be the class index, \( H_i(x) \) be the trained classifier, and \( \varepsilon_i \) be the training error. The final three-class classification is obtained as

\[
H(x) = \begin{cases} 
\arg \max_{i \in C_3} (1 - \varepsilon_i)H_i(x) & \text{if } \exists i \in C_3, H_i(x) > 0 \\
0 & \text{otherwise}
\end{cases}
\]

where the return value 0 represents the failure of the classification. Finally, votes from blocks excluding SHADOW regions determine the class of current terrain. The blocks classified as SHADOW will not be used for the interest point detection since the shadow of a robot can be significant outliers for motion estimation.

The advantage of using AdaBoost is learning efficiency and robustness. In the AdaBoost framework, a strong classifier is generated by the combination of weak classifiers. The training of weak classifiers does not require a large number of training data due to the repeated learning mechanism. Furthermore, the combination of multiple classifiers can improve robustness for different types of terrain. These characteristics of AdaBoost will help operations on unexplored
planets where the number of training images is limited. However, it should be noted that training images should be selected appropriately since AdaBoost may occasionally be prone to overfitting.

3.2 Adaptive Detector Selection

After the texture assessment, the suitable feature detector for the current terrain is selected from a detector set. Under the assumption that the terrain can be classified into two classes ROUGH and SMOOTH, the simplest detector set can be composed of two detectors: one for ROUGH terrains and one for SMOOTH terrains. Let $D_A$ be a detector that is fast and applicable to all ROUGH terrains, and $D_B$ be a stable detector that is tuned to detect enough number of points to estimate correct motion even in SMOOTH terrains. $D_A$ and $D_B$ should be selected based on the detector properties shown in Table 1 or generated by tuning parameters.

The simplest rule to select an appropriate detector is selecting $D_A$ for all ROUGH terrains and $D_B$ for all SMOOTH terrains. Occasionally this rule works well, while it overlooks instability of texture assessment. Suppose a robot is in a mixed terrain of ROUGH and SMOOTH, the
result of texture assessment can differ frame by frame. In such an environment, the VO based on the recent SfM scheme has to change the detectors in every frame, which forces re-extraction of interest points from the previous frame. This is because the interest points can be matched only if they have the same ‘interest’, otherwise there is no guarantee that the same points are detected as interest points.

One approach to avoid the excessive switching is to use a saturating up-down counter that is used for the local branch prediction in the field of computer architecture. This technique is simple but known to generate good estimation with less time and memory usage. The $n$-bit saturating counter is composed of a finite state machine with $2^n$ states (see Figure 2). The counter increments its count if the current terrain is classified as ROUGH, and decrements if it is classified as SMOOTH. The saturating property means the further increment (decrement) has no effect when the counter reaches to its maximum (minimum) value. The detector is selected depending on the highest bit of the counter (0/1). Interestingly, as will be shown in Section 5, the 2-bit saturating counter is the best predictor in this application, which is also claimed to be the best in the local branch prediction [32]. Even so, the number of bits $n$ can be adjusted considering the type of explored environments for the best performance.

4. Experimental Setup

4.1 Datasets

Field experiments were conducted to collect datasets which include i) stereo image pairs, ii) ground truth of position $(x, y, z)$ from Post-Processed Kinematic (PPK) GPS, and iii) ground truth of rotation $(\text{roll}, \text{pitch}, \text{yaw})$ from Attitude and Heading Reference System (AHRS). The experimental places were two volcanic areas in Japan: Urasabaku Desert in Izu-Oshima and Suna senri Desert in Mt. Aso. The terrains covered with volcanic products form natural scenes (see Figure 4). The robots used in the experiments are shown in Figure 3. These robots are the experimental rovers developed by JAXA, which have the capability of semi-autonomous navigation.

The rovers were equipped with stereo camera rigs. Table 2 details the specification of the vision systems. Due to the hardware limitation in communication and scheduling, the systems have low frame rate. However, this does not cause an important issue since the traversing speed of these rovers are slow (approximately $0.1 \text{m/sec}$). The image dataset is composed of more than 1000 stereo image pairs in total, obtained during a long 100m drive and several short drives.

<table>
<thead>
<tr>
<th></th>
<th>Micro-6</th>
<th>Cuatro</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOV [deg]</td>
<td>$40 \times 30$</td>
<td>$87 \times 65$</td>
</tr>
<tr>
<td>Resolution</td>
<td>$320 \times 240$</td>
<td>$640 \times 400$</td>
</tr>
<tr>
<td>Frame rate [Hz]</td>
<td>0.25</td>
<td>0.69</td>
</tr>
<tr>
<td>Baseline [m]</td>
<td>0.270</td>
<td>0.475</td>
</tr>
<tr>
<td>Height [m]</td>
<td>1.450</td>
<td>0.770</td>
</tr>
</tbody>
</table>

Figure 3. Appearance of experimental rovers

Table 2. Camera specification of experimental rovers
4.2 Implementation

The current VO system employs the feature-based SfM algorithm. The algorithm of interest point detection is the hybrid detector described in Section 3. In the texture assessment, the linear combination of statistical values is used as a weak classifier, which is obtained as follows:

1. The four inputs are computed from the probability distribution function $P$ of pixel brightness $m$ for each patch. The inputs are defined as the mean $\mu$, variance $\sigma^2$, energy $E$ and the probability of outliers $p_o$, which are computed as

$$
\mu = \sum_m m P(m), \quad \sigma^2 = \sum_m (m - \mu)^2 P(m),
$$
$$
E = \sum_m P(m)^2, \quad p_o = \sum_{m < \mu - 3\sigma, \mu + 3\sigma < m} P(m).
$$

These four inputs compose the four dimensional input $x$.

2. The weak classifier $h_t(x)$ is generated as the linear combination of four inputs

$$
h_t(x) = p \sum_{i=1}^{4} w_i x_i, \quad p \in \{-1, 1\}
$$

with $w_i$ and $p$ that minimize the estimation error.

The interest point detectors used in the comparative study are Harris [19], Shi-Tomasi [21], DoG [14], FAST [23], Fast Hessian [25], STAR [16]. The STAR detector is an implementation based on the CenSurE detector with some modification for increasing speed and stability. The reasons for this selection is that these detectors are considered to be state-of-the-art, and are widely used in state-of-the-art VO systems. The hybrid detector is composed of two detectors from this detector set. The pair is selected by the performance evaluation described in the next section.

After the detection, features are tracked between consecutive images by feature matching using $7 \times 7$ image patches. Similarity is computed by the normalized correlation between the patches. To improve the matching accuracy, the searching window is limited to 15% of the image size, which is decided heuristically. Moreover, the mutual consistency check is performed to ensure the paring of features in two images.

As a measure of motion estimation, RANSAC (RAndom SAmpLe Concensus) [33] is employed, which has been established as the standard method to estimate motion models from dataset with outliers. The idea is to compute hypotheses from randomly sampled data and then verify the hypotheses using the other sets of data. The hypothesis which achieves the highest consensus with other data is selected as a solution. The implemented RANSAC returns the solution after
Table 3. Number of samples in training and testing datasets

<table>
<thead>
<tr>
<th>Class</th>
<th>Urasabaku (Micro-6)</th>
<th>Sunasenri (Cuatro)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#training</td>
<td>#testing</td>
</tr>
<tr>
<td>ROUGH</td>
<td>292</td>
<td>118</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>626</td>
<td>100</td>
</tr>
<tr>
<td>SHADOW</td>
<td>174</td>
<td>22</td>
</tr>
</tbody>
</table>

100 iterations.

The overall system is implemented in C++. The implementations of the detectors are provided with the OpenCV library\(^1\). While the implementations in OpenCV are slightly different from the original implementations of these detectors, this open source library is integrated in many useful robot middlewares such as ROS\(^2\) and RT-middleware\(^3\) and thereby available on many platforms. It is important to compare with OpenCV’s implementation for the usability evaluation in VO applications.

Execution time is measured only for the core part of algorithms. Reporting time is computed on a Linux machine on Intel Core 2 Quad 2667MHz CPU for an 320×240 grayscale image.

5. Results and Analysis

5.1 Texture Assessment

This experiment evaluates the performance of texture assessment. The dataset to train and verify AdaBoost classifiers are generated from both Urasabaku and Sunasenri datasets by hand-labeling. The detail of the datasets is in Table 3. The SHADOW class in the Sunasenri dataset does not exist since the dataset was collected in a cloudy day.

The ability to classify terrain depends on the off-line learning of AdaBoost classifiers. Three classifiers are trained for classifying input images to a specific class or not. Each classifier uses the images of the specific class as positive training data, and the others as negative examples. For instance, the SMOOTH classifier for the Urasabaku dataset uses 626 SMOOTH images as positive inputs, and 466 images as negative inputs (see Table 3).

Figure 5 shows the learning curves for binary classifiers in the Urasabaku dataset up to 100 weak classifiers. Every classifier achieves 15%-20% correct classification accuracy with 20-30 weak classifiers. According to the learning curves, the number of weak classifiers to compose a strong binary classifier is decided as shown in Table 4. The generalization errors are also stored and used for weighting the classifiers afterward. The classification accuracy of binary classifiers is relatively low in the current implementation, but it will be increased by replacing the current inputs with textons for example. As the main focus of this work is not the classification of terrain and the authors regard the computational efficiency as more important in this step (0.38 ms for one image classification), the accurate classification is left as a subject for future works.

The combination of three binary classifiers generates the final hypothesis. Table 5 shows the confusion matrices of two datasets. The AdaBoost-based classifiers give the accuracy of 75.8% and 68.3% for the Urasabaku and Sunasenri datasets, respectively. The remarkable confusion in the table is that the classifiers regard ROUGH terrains as SMOOTH terrains. Several reasons can be found for this, such as inaccuracy of hand-labeling and the weighting proportion of binary classifiers (the SMOOTH classifier has lower error rate than the ROUGH classifier). Analysis on these topics may help improving the classification.

\(^1\)http://sourceforge.net/projects/opencvlibrary
\(^2\)http://www.ros.org/wiki
\(^3\)OpenRTM-aist(http://openrtm.org/openrtm/en/content/download)
Figure 5. Learning curves of binary classifiers in Urasabaku dataset

Table 4. Classification accuracy for Urasabaku and Sunasenri datasets

<table>
<thead>
<tr>
<th>Class</th>
<th>Urasabaku (Micro-6)</th>
<th>Sunasenri (Cuatro)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#classifiers</td>
<td>%error</td>
</tr>
<tr>
<td>ROUGH</td>
<td>25</td>
<td>21.2</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>11</td>
<td>14.2</td>
</tr>
<tr>
<td>SHADOW</td>
<td>10</td>
<td>17.1</td>
</tr>
</tbody>
</table>

Table 5. Confusion matrices of patch classification in Urasabaku and Sunasenri datasets

<table>
<thead>
<tr>
<th>Actual</th>
<th>Prediction</th>
<th>ROUGH</th>
<th>SMOOTH</th>
<th>SHADOW</th>
<th>N/A</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urasabaku</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75.8</td>
</tr>
<tr>
<td>ROUGH</td>
<td>82</td>
<td>20</td>
<td>6</td>
<td>10</td>
<td>69.5</td>
<td></td>
</tr>
<tr>
<td>SMOOTH</td>
<td>2</td>
<td>91</td>
<td>3</td>
<td>4</td>
<td>91.0</td>
<td></td>
</tr>
<tr>
<td>SHADOW</td>
<td>5</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>40.9</td>
<td></td>
</tr>
<tr>
<td>Sunasenri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68.3</td>
</tr>
<tr>
<td>ROUGH</td>
<td>44</td>
<td>21</td>
<td>-</td>
<td>4</td>
<td>63.8</td>
<td></td>
</tr>
<tr>
<td>SMOOTH</td>
<td>12</td>
<td>38</td>
<td>-</td>
<td>1</td>
<td>74.5</td>
<td></td>
</tr>
</tbody>
</table>

5.2 Saturating Counter

As for the saturating counter of hybrid detector, the combination of detectors and the size of saturating counter is evaluated. The dataset for the evaluation is composed of 962 stereo pairs in the long drive of Urasabaku dataset.

According to the recent evaluation results of interest point detectors described in Section 2.2, three detectors (Harris, FAST, and DoG) are chosen and combined to form hybrid detectors. The performance is compared in Figure 6 in terms of the effective percentage of correct matches and the processing time including the switching delay. Figure 6(a) shows a statistical result of repeatability, presenting the sorted percentage of correct matches for every frame that includes more than 20 matches. Figure 6(b) shows the average processing time for a frame. One can find the effectiveness of the combination of Harris and FAST detectors: four times faster, highly stable as well as having higher matching rate. Therefore, Harris and FAST detectors are employed into the hybrid detector.

The number of states in the saturating counter is another important parameter. While the larger counter improves the stability against local terrain changes and classification errors, the smaller counter gives a quick adaptiveness to terrain changes. The experiment was conducted for 1-bit, 2-bit, and 3-bit saturating counters. Note that the 1-bit saturating counter implements the simplest selection rule, i.e., the classification result directly determines a detector.

The results in Figure 7 shows that the saturating counters with bits greater than two are slightly better in terms of stability than the simplest selection. Moreover, the timing analysis gives the further advantage of the saturating counter. Specifically, the saturating counter saves more than 30% of the detection time in total, because the branch prediction of saturating counter
performs the rejection of classification deviations. For simplicity, the 2-bit saturating counter is adopted.

5.3 Comparison of Detectors

This experiment compares the proposed method with commonly used detectors (Harris, Shi-Tomasi, FAST, DoG, Fast Hessian, and STAR). As a measure of repeatability, Figure 8 presents how many of the detected points are correctly matched for a given level of matching quality. While the motion of a calibrated camera with six degrees of freedom (6DoF) is computed from a minimum of five point correspondences in theory, 10-30 matches are typically used for improving the robustness [4, 16, 34]. A good detector should have high repeatability (vertical axis) as well as high stability to all types of terrain (horizontal axis). The proposed detector which is composed of Harris and FAST successfully benefits from both detectors, and outperforms the other detectors.

Figure 9 shows the percentage of frames with correct matches less than $N$. It represents how many frames will fail if a threshold for the number of inliers is given to assure the reliability of motion estimation. Harris, Shi-Tomasi and the proposed detector have less than 10% missed frames with minimum of 20 inliers even for the extremely untextured dataset.

The VO estimates appear in Figure 10. Figure 10(a) shows trajectories of the rover with respect to different detectors, a trajectory estimated from the wheel encoders, and ground truth by PPK-GPS. DoG exhibits the best 2D position estimate, and Harris and the proposed detector follow. FAST and STAR perform poorly in the current setups. In fact, as shown in Figure 9, these detectors could not provide enough number of inliers to estimate the motion in certain terrains. Figure 10(b) shows the RMS errors in the 2D field. The VO system with the hybrid detector drifts 8% within 50m driving.

The result of processing time is reported in Table 6. The proposed method could detect interest points from an image with 9.64 msec, which is 20% acceleration of single Harris. This result can vary with dataset; for example, the proposed method would perform better for a dataset...
Figure 8. Statistical results on stability and accuracy: the sorted percentage of correct matches for minimum of \( N \) match frames

Figure 9. Missed frames: the percentage of frames as a function of the number of correct matches containing ROUGH images at a high rate since the high-speed detector will be selected in most images. The dataset used in the experiment contains approximately 30% ROUGH terrains.

5.4 Further Analysis

Although the proposed detection scheme robustly detects interest points from a wide range of terrain types, motion estimation failures can occasionally occur if an image is in extremely bad conditions. An example that was found in the datasets is lens flare which occurs if the sun-light strikes the lens and is reflected onto the film. Such undesirable effects is hard to avoid when an experiment is conducted in a sunny day. Such problems related to the outdoor illumination conditions should be solved by means of camera deployment or operational effort.

In the case where image conditions are well enough, the switching scheme allows wider range of terrain types. Even so, it is hard to achieve perfect motion estimation. VO inevitably suffers from accumulated errors as shown in Figure 10(b). The off-road VO specifically accumulates errors, since such environments typically have a low-textured scene with a small high contrast element and hence the poor distribution of tracked points spoils the accuracy of motion estimation. This
Table 6. Comparison of processing time of interest point detectors for a 320x240 image on Intel Core 2 Quad 2667MHz CPU. The time of proposed method includes terrain classification.

<table>
<thead>
<tr>
<th>Detector</th>
<th>Harris</th>
<th>Shi-Tomasi</th>
<th>FAST</th>
<th>DoG</th>
<th>Fast Hessian</th>
<th>STAR</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proc. time [ms]</td>
<td>11.87</td>
<td>14.36</td>
<td>1.32</td>
<td>54.98</td>
<td>27.90</td>
<td>9.86</td>
<td>9.64</td>
</tr>
</tbody>
</table>

effect can be observed independently of the kind of interest point detectors. Several approaches have attacked this problem and found solutions such as bundle adjustment and loop closing. VO in outdoor environments should follow these techniques in order to enable long-range reliable estimation.

6. Conclusion and Future Works

This paper presented a novel algorithm that enables robust and fast interest point detection in untextured natural environments. The algorithm performs vision-based terrain classification, and dynamic terrain-adaptive selection of interest point detectors. The experiments show that the technique can robustly detect interest points in both featureless and rich terrains without decreasing the computational speed. The development of this technique will help VO in outdoor scenery, and thereby spread applications of outdoor mobile robots.

Several improvements can be suggested. Firstly, it would be better to improve the accuracy of terrain classification without increasing the processing time. Despite the ability of the saturating counter to mitigate misclassification, the accurate terrain classification reduces unnecessary switching and can also be extended to further strategy. Secondly, due to limitations of this work the estimation error is accumulated during long range traverses. The accumulated error may be decreased by previously known techniques.

In general, it is hard to estimate terrain conditions without preliminary surveys. The terrain-
adaptive strategy mentioned in this paper makes the system more robust since it allows the system to have several assumptions and to prepare multiple detectors suitable for variant terrains. Hopefully, this technique can help the robustness of vision-based algorithms and be used in various applications of autonomous mobile robots.

Appendix A. Parameters of Interest Point Detectors

Table A1 shows the parameter values used in the comparative study of interest point detectors.

### Table A1. Parameter values used for the interest point detectors

<table>
<thead>
<tr>
<th>Detector</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris</td>
<td>k</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>threshold</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>non-maximum</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>suppression radius</td>
<td>5</td>
</tr>
<tr>
<td>Shi-Tomasi</td>
<td>threshold</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>non-maximum</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>suppression radius</td>
<td>5</td>
</tr>
<tr>
<td>FAST</td>
<td>threshold</td>
<td>80</td>
</tr>
<tr>
<td>Fast Hessian</td>
<td>octaves</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>layers per octave</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>threshold</td>
<td>400</td>
</tr>
<tr>
<td>DoG</td>
<td>octaves</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>levels per octave</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>σ₀</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>θ_edge</td>
<td>10</td>
</tr>
<tr>
<td>STAR</td>
<td>θ_contr</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>max scale</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>response threshold</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>line threshold (binarized)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>line threshold (projected)</td>
<td>8</td>
</tr>
</tbody>
</table>

References