Image Features and Seasons Revisited

Tomáš Krajník[1] Pablo Cristóforis[2] Matías Nitsche[2] Keerthy Kusumam[1] Tom Duckett[1]

***Abstract*— We present an evaluation of standard image features in the context of long-term visual teach-and-repeat mobile robot navigation, where the environment exhibits significant changes in appearance caused by seasonal weather variations and daily illumination changes. We argue that in the given long-term scenario, the viewpoint, scale and rotation invariance of the standard feature extractors is less important than their robustness to the mid- and long-term environment appearance changes. Therefore, we focus our evaluation on the robustness of image registration to variable lighting and naturally-occurring seasonal changes. We evaluate the image feature extractors on three datasets collected by mobile robots in two different outdoor environments over the course of one year. Based on this analysis, we propose a novel feature descriptor based on a combination of evolutionary algorithms and Binary Robust Independent Elementary Features, which we call GRIEF (Generated BRIEF). In terms of robustness to seasonal changes, the GRIEF feature descriptor outperforms the other ones while being computationally more efficient.**

***Index Terms*— visual navigation, mobile robotics, long-term autonomy**

I. Introduction

Cameras are becoming a de-facto standard in sensory equipment for mobile robotic systems including field robots. While being affordable, small and light, they can provide high resolution data in real time and virtually unlimited measurement ranges. Moreover, they are passive and do not pose any interference problems even when deployed in the same environment in large numbers. Most importantly, the computational requirements of most machine vision techniques are no longer a significant issue due to the availability of powerful computational hardware. Hence, onboard cameras are often used as the primary sensors to gather information about the robot's surroundings.

Many visual robot navigation and visual SLAM methods rely on local image features [1] that allow to create sparse, but information-rich image descriptions. These methods consist of a detection and a description step, which extract salient points from the captured images and describe the local neighborhood of the detected points. Local features are meant to be repeatably detected in a sequence of images and matched using their descriptors, despite variations in the viewpoint or illumination. Regarding the quality of feature extractors, a key paper of Mikolajczyk ans Schmid [2] introduced a methodology for evaluation of feature invariance to image scale, rotation, exposure and camera viewpoint changes. Mukherjee et al. [3] evaluated a wide range of image feature detectors and descriptors, confirming the superior performance of the SIFT algorithm [4]. Other comparisons were aimed at the quality of features for visual odometry [5] or visual Simultaneous Localization and Mapping (SLAM) [6]. Unlike the aforementioned works, we focus our evaluation on navigational aspects, especially to achieve long-term autonomy under seasonal changes.

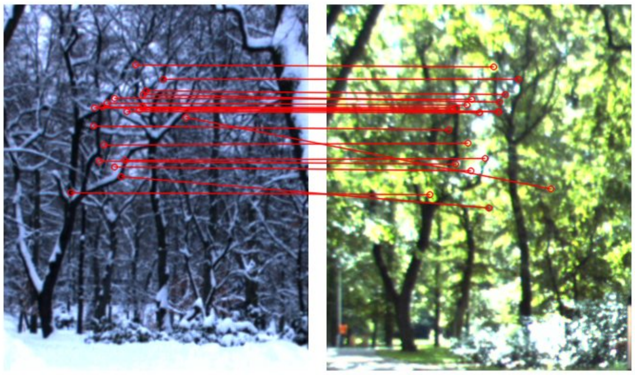


Fig. 1: Examples of tentative matches of the GRIEF image features across seasonal changes.

The problem of vision-based long-term autonomous localization has been addressed by several researchers. Dayoub and Duckett [7] proposed a method where the robot continuously adapts its environment model by identifying stable image features and forgetting the unstable ones. Milford and Fraser [8] proposed to use image sequences to identify places across seasons, while Churchill and Newman [9] clustered different observations of the same place to form "experiences" that characterize the place appearance in particular conditions. Neubert et al. [10] and Krajnik et al. [11] proposed to learn how the environment changes over time and used the learned model to predict the likely appearance of the places at a given time. Another approach uses dead reckoning to predict which place the vehicle is close to, loads a bank of Support Vector Machine classifiers associated with that place and uses these to obtain a metric pose estimate [12]. Carlevaris et al. proposed to learn visual features that are robust to the appearance changes and showed that the learned features outperform the SIFT and SURF feature extractors [13]. Other approaches used the visual information to guide the robot along a given path without performing localization. For example, [14] reported that the combination

[1] Lincoln Centre for Autonomous Systems, University of Lincoln, UK {tkrajnik,kkusumam,tduckett}@lincoln.ac.uk

[2] Laboratory of Robotics and Embedded Systems, Faculty of Exact and Natural Sciences, University of Buenos Aires {pdecris,mnitsche}@dc.uba.ar

The work has been supported by the EU ICT project 600623 'STRANDS' and UBACYT project 20020130300035BA. We would like to thank Nicholas Carlevaris-Bianco for sharing the Michigan dataset.

978-1-4673-9163-4/15/$31.00 © 2015 IEEE

of an entropy-based map selection and histogram voting scheme to correct the robot heading allows to use three-year-old maps for autonomous outdoor navigation.

Let us consider a mobile robot navigating along a known (previously mapped) path in an outdoor environment. In this case, it is not necessary to use image features that are highly invariant to large viewpoint changes since the robot keeps itself close to the intended path. The rotational invariance is also not crucial for navigation, because one can assume that the robot moves on a locally planar surface. On the other hand, the map provided to the robot might be obsolete, because the environment appearance changes over time [15] due to illumination variations, weather conditions and seasonal factors.

These considerations about the visual features motivate us to analyze available feature detector and descriptor algorithms in terms of their long-term performance in autonomous navigation based on a the teach-and-repeat principle, e.g., as used in [16], [17], [18], [19].

First, we present our proposed evaluation methodology and the results achieved using recent feature detectors and descriptors such as BRIEF, (root)SIFT, ORB and BRISK, which are freely available as a part of open source implementations. Second, we propose a new feature descriptor based on evolutionary methods and binary comparison tests. This algorithm, called GRIEF (Generated BRIEF), outperforms the aforementioned image feature extractors in terms of its ability to deal with naturally-occurring seasonal changes and lighting variations. This adaptive approach allows to automatically generate visual feature descriptors that are more robust to environment changes than standard hand-designed features.

II. Local image feature extractors

Local image features provide a sparse, but distinctive representation of images so that these can be retrieved, matched or registered efficiently. The feature extraction process consists of two successive phases: feature detection and feature description. The feature detector identifies a salient area in an image, e.g. a corner, blob or edge, which is treated as a keypoint. The feature descriptor creates a vector that characterizes the neighborhood of the detected keypoint, typically in a scale-affine invariant way. Typical descriptors capture various properties of the image region like texture, edges, intensity gradients, etc.

The features are meant to be repeatably extracted from different images of the same scene even under conditions of unstable illumination or changing viewpoints. In this paper, we evaluate six popular and one novel local image feature extraction algorithms for the purpose of long-term robot navigation. Six of these algorithms are included in the Open Source Computer Vision (OpenCV) software library (version 2.4.3), which was used to generate the results presented in this paper.

A. Scale Invariant Feature Transform - SIFT

The Scale Invariant Feature Transform (SIFT) is probably the most popular local feature extractor [4] due to its scale and rotation invariance and robustness to lighting and viewpoint variations. It combines a Difference-of-Gaussian (DoG) detector and a descriptor based on gradient orientation histograms. The feature detection process first generates a scale space of the image by convolving it with Gaussian kernels of different sizes. The DoG detector then searches for local extrema in the images obtained by the difference of two adjacent scales in the Gaussian image pyramid. This gives an approximation of the Laplacian of Gaussian (LoG) function where local extrema correspond to locations of blob-like structures. This type of keypoint localization allows to detect blobs at multiple scales, resulting in scale invariance of the features. To achieve rotation invariance, SIFT assigns a dominant orientation to the detected keypoint obtained by binning the gradient orientations of its neighborhood pixels. Then, the SIFT descriptor is formed by sampling the image gradient magnitudes and orientations of the region around the keypoint while taking into account the scale and rotation calculated in the previous steps. While being precise, distinctive and repeatable, calculation of the SIFT feature extractor is computationally demanding. Arandjelović and Zisserman [20] showed that a simple normalization (called Root-SIFT) improves SIFT performance in object retrieval scenarios.

B. Speeded Up Robust Features - SURF

Inspired by SIFT, the Speeded Up Robust Feature (SURF) extractor was first introduced by Bay et al. [21]. The main advantage of SURF is its speed - the experiments presented in [21] show that it is significantly faster than SIFT with no considerable performance drop in terms of invariance to viewpoint, rotation and scale changes. The speedup is achieved through the use of integral images that allow to calculate the response of arbitrarily-sized 2D box filters in constant time. The box filters are used both in the detection step, where they approximate the Hessian determinant, and in the description phase for spatial binning similar to SIFT. The (rather inefficient) rotation estimation step can be omitted from the SURF algorithm, resulting in 'Upright SURF', that is not rotation invariant. This might be beneficial in some applications, for example Valgren and Lilienthal [15] showed that U-SURF outperforms SURF in long-term outdoor localization.

C. STAR feature detector

The STAR feature detector is a variant of the Centre Surround Extrema (CenSurE) detector [22]. The authors of CenSurE argue that the keypoint localization precision of the multi-scale detectors like SIFT and SURF becomes low because of the interpolation used at higher levels of the scale space. The CenSurE detector circumvents this issue as it searches for keypoints as extremas of the centre surround filters at multiple scales. Thus, the scale space is generated by using masks of different sizes rather than using interpolation, which has a negative impact on detection precision. While CenSurE uses polygons to approximate the circular filter mask, the STAR feature approximates it by using two square masks (one upright and one rotated at 45 degrees). Similarly to SURF, this scheme allows for efficient box filter response calculation at multiple scales, resulting in the computational efficiency of STAR.

D. Binary Robust Independent Elementary Features - BRIEF

The BRIEF feature descriptor uses binary strings as features, which makes its construction, matching and storage highly efficient [23]. The binary string is computed by using pairwise comparisons between pixel intensities in an image patch that is first smoothed by a Gaussian kernel to suppress noise. In particular, the value of the i^{th} bit in the string is set to 1 if the intensity value of a pixel in position x_i, y_i is greater than the intensity of a pixel at position x'_i, y'_i. Since the sequence of test locations of the comparisons $\delta_i = (x_i, y_i, x'_i, y'_i)$ can be chosen arbitrarily, the authors of [23] compared several schemes for generating δ_i and determined the best distribution to draw δ_i from. The binary strings are matched using Hamming distance, which is faster than using the Euclidean distance as in SIFT or SURF. In [23], the authors consider 128, 256 and 512 binary string sizes referred to as BRIEF-16, BRIEF-32, BRIEF-64 respectively.

E. Features from Accelerated Segment Test - FAST

The FAST detector compares intensities of pixels lying on a 7-pixel diameter circle to the brightness of the circle's central pixel. The 16 pixels of the circle are first marked as bright, neutral or dark depending on their brightness relative to the central pixel. The central pixel is considered a keypoint if the circle contains a contiguous sequence of at least n bright or dark pixels (a typical value of n is 12). In order to quickly reject candidate edges, the detector uses an iterative scheme to sample the circle's pixels [24]. For example, the first two examined pixels are the top and bottom one - if they do not have the same brightness, a contiguous sequence of 12 pixels cannot exist and the candidate edge is rejected. This fast rejection scheme causes the FAST detector to be computationally efficient. In [3], the combination of FAST detector and SIFT descriptor shows performance similar to original SIFT, while being faster to calculate.

F. Oriented FAST and Rotated BRIEF - ORB

The ORB feature extractor combines a FAST detector with orientation component (called oFAST) and a steered BRIEF (rBRIEF) descriptor. The goal of ORB is to obtain robust, fast and rotation invariant image features meant for object recognition and structure-from-motion applications. The keypoints are identified by the FAST detector and ordered by the Harris corner measure at multiple scales [25]. Then, the orientation of the feature is calculated using the intensity centroid. The coordinates of the pair of points for comparison are rotated according to this value and the comparisons are then performed. However, the rotation invariance introduced in ORB has a negative impact on its distinctiveness. Thus, the authors of ORB employed machine learning techniques to generate the comparison points so that the variance of the comparisons would be maximized and their correlation minimized.

G. Binary Robust Invariant Scalable Keypoints - BRISK

The BRISK feature detector is scale and rotation invariant [26]. To identify the keypoint locations, BRISK uses the AGAST [27] feature detector which is an accelerated variant of FAST. The scale invariance of BRISK is achieved by detecting keypoints on a scale pyramid [26]. The descriptor of BRISK is a binary string that is based on binary point-wise brightness comparisons similar to BRIEF. Unlike BRIEF or ORB, which use a random or learned comparison pattern, BRISK's comparison pattern is centrally symmetric. While the outermost points of the comparison pattern are used to determine the feature orientation, the comparisons of the inner points form the BRISK binary descriptor.

III. GRIEF: GENERATED BRIEF SEQUENCE

The standard BRIEF descriptor is a binary string that is calculated by 256 intensity comparisons of pixels in a 48×48 image region surrounding the keypoint provided by a detector. In principle, the locations of the pixel pairs to be compared can be chosen arbitrarily, but they have to remain static after the choice has been made. Realizing that the choice of the comparison locations determines the descriptor performance, the authors of BRIEF and ORB attempted to find the best comparison sequences. While the authors of the original BRIEF proposed to select the sequences randomly from a two-dimensional Gaussian distribution, the authors of ORB chosen the locations so that the variance of the comparisons is high, but their correlation is low.

Given an image $\mathbf{I}$, a BRIEF descriptor $\mathbf{b}(\mathbf{I}, c_x, c_y)$ of an interest point c_x, c_y (detected by the STAR algorithm) is a vector consisting of 256 binary numbers $b_i(\mathbf{I}, c_x, c_y)$ calculated as

$$b_i(\mathbf{I}, c_x, c_y) = \mathbf{I}(x_i + c_x, y_i + c_y) > \mathbf{I_j}(x'_i + c_x, y'_i + c_y). \quad (1)$$

Since the position c_x, c_y is provided by the feature detector, the BRIEF descriptor calculation is defined by a sequence Δ of 256 vectors $\delta_\mathbf{i} = (x_i, y_i, x'_i, y'_i)$ that define pixel positions for the individual comparisons. Thus, the BRIEF method calculates the dissimilarity of interest point $\mathbf{a}$ with coordinates (a_x, a_y) in image $\mathbf{I_a}$ and interest point $\mathbf{b}$ with coordinates (b_x, b_y) in image $\mathbf{I_b}$ by the Hamming distance of their binary descriptor vectors $\mathbf{b}(\mathbf{I_a}, a_x, a_y)$ and $\mathbf{b}(\mathbf{I_b}, b_x, b_y)$. Formally, the dissimilarity $d(\mathbf{a}, \mathbf{b})$ between points $\mathbf{a}$ and $\mathbf{b}$ is

$$d(\mathbf{a}, \mathbf{b}) = \sum_{i=0}^{255} d_i(\mathbf{a}, \mathbf{b}), \quad (2)$$

where $d_i(\mathbf{a}, \mathbf{b})$ are the differences of the individual comparisons δ_i calculated as

$$d_i(\mathbf{a}, \mathbf{b}) = |b_i(\mathbf{I_a}, a_x, a_y) - b_i(\mathbf{I_b}, b_x, b_y)|. \quad (3)$$

Let us assume that the BRIEF method has been used to establish tentative correspondences of points in two images, producing a set $\mathcal{P}$ of point pairs $\mathbf{p_k} = (\mathbf{a_k}, \mathbf{b_k})$. Now, let us assume that the tentative correspondences were marked as either 'correct' or 'false', e.g. by RANSAC-based geometrical verification [28]. This allows to split $\mathcal{P}$ into a set of correct correspondence pairs $\mathcal{P}_C$ and a set of incorrectly established pairs $\mathcal{P}_F$. This allows to calculate the fitness $f(\delta_i, \mathcal{P}_C, \mathcal{P}_F)$ of each individual comparison δ_i as

$$f(\delta_i, \mathcal{P}_C, \mathcal{P}_F) = \sum_{\mathbf{p} \in \mathcal{P}_C} (1 - 2\, d_i(\mathbf{p})) + \sum_{\mathbf{p} \in \mathcal{P}_F} (2\, d_i(\mathbf{p}) - 1). \tag{4}$$

The first term of Equation (4) penalizes the comparisons δ_i that increase the Hamming distance of correctly established correspondences and increases the fitness of comparisons that do not contribute to the Hamming distance. The second term of Equation (4) improves the fitness of comparisons that indicate the differences of incorrectly established correspondences, while penalizing those comparisons that do not increase the Hamming distance. The fitness function $f(\delta_i)$ allows to rank the comparisons according to their contribution to the descriptor's distinctiveness.

The sets $\mathcal{P}_C$ and $\mathcal{P}_F$, which serve as positive and negative training samples, can contain correspondences from several image pairs, which allows to calculate the fitness $f(\delta_i)$ for larger datasets. The fitness evaluation of the individual components (comparisons) of the descriptor allows to train GRIEF for a given dataset through an iterative procedure that repeatedly evaluates the contribution of individual comparisons δ_i to the feature's distinctiveness and substitutes the 'weak' comparisons by random vectors, see Algorithm 1.

At first, the training method extracts positions of the interest points of all training images and calculates the descriptors of these keypoints using the latest comparison sequence Δ. Then, it establishes tentative correspondences between the extracted keypoints and uses geometric constraints characteristic for the robot movement [18] to determine which of the tentative correspondences are correct (these are added to $\mathcal{P}_C$) and which are false (these are added to set $\mathcal{P}_F$). After that, it uses Equation (4) to rank the individual pixel-wise comparisons δ_i. Then, it discards the 10 comparisons with the lowest fitness and generates new ones by drawing (x_i, y_i, x'_i, y'_i) from a uniform distribution. The aforementioned procedure is repeated as long as the overall fitness $f_\Delta = \sum f(\delta_i)$ grows. The resulting comparison sequence Δ is better tuned for the given dataset.

We have trained the GRIEF descriptor on 12 images captured during different months at the first location of the Stromovka dataset, see Figure 4a. The evolution of the GRIEF fitness during the training procedure and the initial (BRIEF) and trained (GRIEF) comparison pairs are shown in Figure 2. Note that except for the locations of pixels to be compared, the working principle of the GRIEF feature is identical to BRIEF and the time for computation and matching is unaffected. The GRIEF sequence training took approximately 1 hour on an i7 machine.

IV. Evaluation datasets

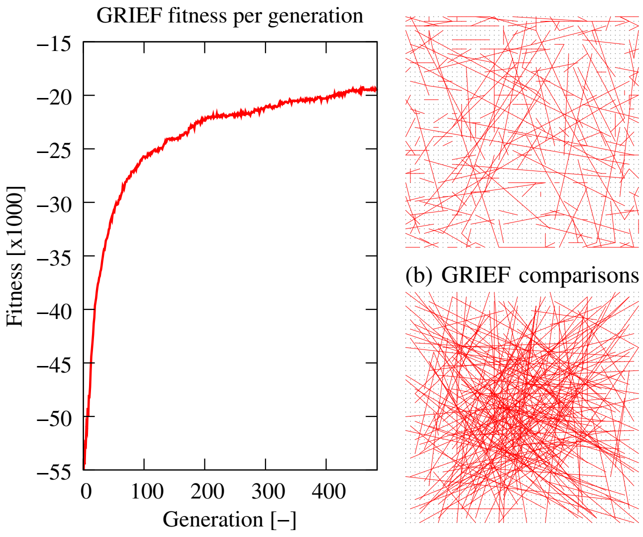
The feature evaluation was performed on three different datasets collected by mobile robots over the course of several months. The first, 'Planetarium' dataset covers seasonal changes in a small forest area near Prague's planetarium in the Czech Republic during the years of 2009 and 2010 [18]. The second 'Stromovka' dataset consists of 2500 images captured during two 1.3 km long tele-operated runs in the Stromovka forest park in Prague during winter and summer 2011. The third 'Michigan' dataset was gathered around the University of Michigan North Campus during 2012 and 2013 [13].

Algorithm 1: GRIEF comparison sequence training

```
Input:  $\mathcal{I}$ – a set of images for GRIEF training,
        $\Delta_0$ – initial comparison sequence – BRIEF
        $f_{min}$ – minimal fitness change (stop condition)
Output: $\Delta$ – improved comparison sequence – GRIEF
// calculate keypoints in all images
foreach $\mathbf{I} \in \mathcal{I}$ do
    $\mathcal{C}_\mathbf{I} \leftarrow$ STAR($\mathbf{I}$)
// start GRIEF training
while $f' > f_{min}$ do
    // extract GRIEF features
    foreach $\mathbf{I} \in \mathcal{I}$ do
        $\mathcal{B}_\mathbf{I} \leftarrow \emptyset$      // clear descriptor set
        foreach $(c_x, c_y) \in \mathcal{C}_\mathbf{I}$ do
            $\mathcal{B}_\mathbf{I} \leftarrow \{\mathcal{B}_\mathbf{I} \cup \text{GRIEF}(\mathbf{I}, c_x, c_y)\}$
    // generate training samples
    $\mathcal{P}_C, \mathcal{P}_F \leftarrow \emptyset$   // initialize sample sets
    foreach $\mathbf{I}, \mathbf{J} \in \mathcal{I}$ do
        // calculate correspondences
        if $\mathbf{I} \neq \mathbf{J}$ then
            // tentative correspondences
            $\mathcal{P} \leftarrow \text{match}\{\mathcal{B}_\mathbf{I}, \mathcal{B}_\mathbf{J}\}$
            // epipolar constraints
            $(\mathcal{P}'_C, \mathcal{P}'_F) \leftarrow$ RANSAC ($\mathcal{P}$)
            // add results to sample sets
            $\mathcal{P}_C \leftarrow \{\mathcal{P}_C \cup \mathcal{P}'_C\}$
            $\mathcal{P}_F \leftarrow \{\mathcal{P}_F \cup \mathcal{P}'_F\}$
    // establish fitness of $\delta_i$ by (4)
    for $i \in 0..255$ do
        $f(\delta_i) \leftarrow \sum_{\mathcal{P}_C} (1 - 2\, d_i(.)) + \sum_{\mathcal{P}_F} (2\, d_i(.) - 1)$
    // establish overall fitness
    $f_\Delta \leftarrow \sum_{i=0}^{255} f(\delta_i)$
    // estimate fitness improvement
    $f' \leftarrow 0.99\, f' + 0.01\, (f_\Delta - f_{last})$
    $f_{last} \leftarrow f_\Delta$
    // replace 10 least-fit comparisons
    for $i \in 0..9$ do
        $\delta_w \leftarrow \arg\min_{\delta \in \Delta}(f(\delta))$   // least fit $\delta$
        $\Delta \leftarrow \{\Delta \setminus \delta_w\}$          // gets replaced
        $\Delta \leftarrow \{\Delta \cup \text{random}\, \delta_i\}$   // by a random $\delta$
```

A. The Planetarium dataset

The Planetarium dataset was obtained by a Unibrain Fire-i601c camera mounted on a P3-AT mobile robot. During the first data collection, the mobile robot was manually driven through a closed path and created a topological-landmark



(a) GRIEF fitness per generation

(b) GRIEF comparisons

(c) BRIEF comparisons

Fig. 2: The evolution of the GRIEF fitness and the initial (BRIEF) and trained (GRIEF) comparison pairs. The GRIEF comparisons favour shorter distances.

map. On the following month, the robot used a robust navigation technique [18] to repeat the same path using the map from the previous month. During the autonomous run, the robot created a new map and recorded images from its on-board camera again. This procedure was repeated every month from September 2009 until the end of 2010, resulting in 16 different image sequences [29].

Although the path started at an identical location every time, the imprecision of the autonomous navigation system caused slight variations in the robot position when traversing the path. Therefore, the first image of each traversed path is taken from exactly the same position, while the positions of the other pictures may vary by up to ±0.8 m.

Although the original data contains thousands of images, we have selected imagery only from 5 different locations in 12 different months, see Figures 3 and 4.

Six independent persons were asked to register the images and to establish their relative horizontal displacement, which corresponds to the relative robot orientation at the times the images were taken. The resulting displacements were checked for outliers (these were removed) and the averaged estimations were used as ground truth.

B. The Michigan dataset

The Michigan dataset (or North Campus Long Term Dataset) was collected by a research team at the University of Michigan for their work on image features for dynamic lighting conditions [13]. The dataset consists of 27 sessions performed over 15 months. Each session contains a sequence of 1232×1616 color images obtained by a Segway robotic platform that was guided around the University of Michigan North Campus. The authors of the dataset provided us with images captured from 5 different locations taken during different sessions. Since this dataset does not seem to be captured on an exactly regular basis and some months were missing, we selected 12 images of each place in a way that would favour their uniform distribution throughout a year. Then, we removed the uppermost and bottom parts of the images that contain ground plane or sky and resized the rest to 1024×386 pixels while maintaining the same aspect ratio, see Figures 5 and 6. The resulting dataset has the same format as the Stromovka one and was evaluated in exactly the same way. However, the ground truth data were established only by one person. Unlike the Stromovka dataset, the Michigan one contains less foliage and more buildings, and therefore is less influenced by the naturally occurring seasonal changes.

C. The Stromovka dataset

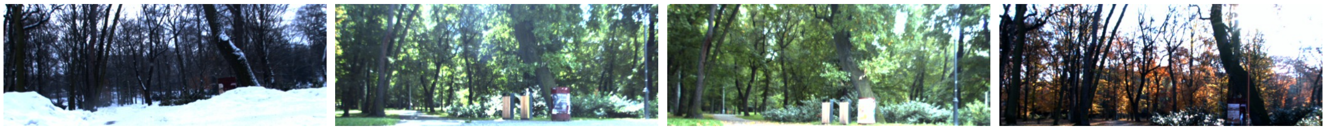
The Stromovka dataset consists of two image sequences captured in winter and summer along a 1.3 km long path through diverse terrain of the Stromovka park in Prague. The appearance of the environment between the two sequences changes significantly (see Figure 7), which makes the Stromovka dataset especially challenging. The magnitude of the appearance change should allow for better evaluation of the feature extractors' robustness to environment variations. The equipment to capture the Stromovka dataset was the same as for the Planetarium one. Unlike the Planetarium dataset, where the robot used a precise navigation technique, the Stromovka data collection was tele-operated and the recorded trajectories are sometimes more than 2 m apart.

V. Evaluation

In our evaluation, we are considering that the robot is using a visual-based teach-and-repeat method that does not require full six degree-of-freedom global localization. Instead, we assume that the teach-and-repeat method uses the visual data to correct the robot's orientation in order to keep it on the path it has been taught [16], [17], [18], [19]. Therefore, we evaluate the feature extraction and matching algorithms in terms of their ability to establish the correct orientation of the robot under environment and lighting variations.

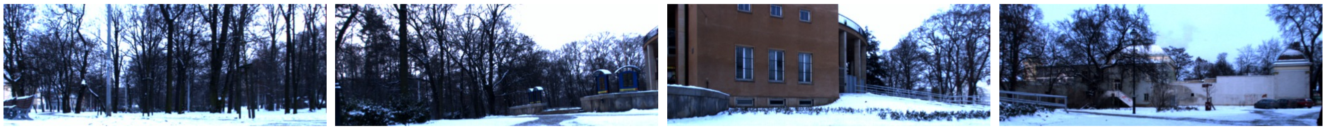
Two methods were considered for determining the relative rotation of the camera. The first method closely follows a classical approach used in computer vision where known camera parameters and correspondences between extracted and mapped features are used to calculate the essential matrix, which is factored to obtain the robot rotation. An alternative method used in [16], [18] calculates a histogram of horizontal (in image coordinates) distances of the tentative correspondences and calculates the robot orientation from the highest-counted bin. In other words, the robot orientation is established from the mode of horizontal distances of the corresponding pairs by means of histogram voting. In all the tests performed, the histogram voting method performed better than the one based on the essential matrix and thus, the latter is not included in the evaluation.

Since the proposed evaluation is based on a measure of the feature extractor's ability to establish the robot heading, we calculate its 'error rate' as the ratio of incorrect to total heading estimates. An orientation estimate is considered as



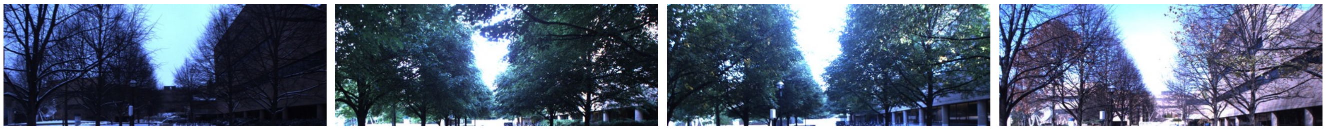
(a) January 2010 (b) May 2010 (c) August 2010 (d) October 2010

Fig. 3: Examples of seasonal changes at the location II of the Planetarium dataset.



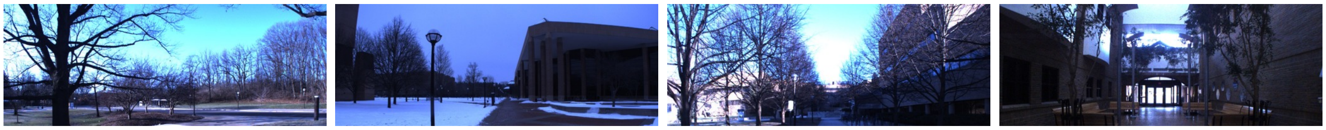
(a) Planetarium - location I (b) Planetarium - location III (c) Planetarium - location IV (d) Planetarium - location V

Fig. 4: View from the robot camera at different locations of the Planetarium dataset.



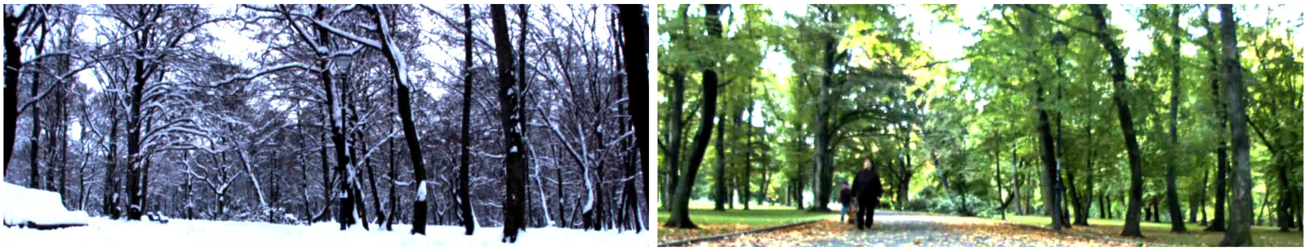
(a) January 2012 (b) May 2010 (c) August 2012 (d) November 2012

Fig. 5: Sample pictures capturing the seasonal changes at location I of the Michigan dataset.



(a) Michigan - location II (b) Michigan - location III (c) Michigan - location IV (d) Michigan - Location V

Fig. 6: View from the robot camera at different locations of the Michigan dataset.



(a) Stromovka winter image example (b) Stromovka summer image example

Fig. 7: View from the robot camera at a single location of the Stromovka dataset.

correct if it differs from the ground truth by less than 20 pixels, which is the width of the bin used by the histogram voting method. The error rate for estimating the correct heading is heavily dependent on the number of extracted features, which depends on the setting of a 'peak threshold' of a particular feature detector. Therefore, we had to establish 16 different peak thresholds for each detector and dataset, such that setting these values will result in detection of $\{100, 200, \ldots, 1600\}$ features per dataset image (on average). The dataset images were processed by the feature extractors with thresholds set to these particular values. The feature correspondences between each pair of the datasets' images from the same location were established by the ratio test[1] method suggested in [4], [23]. Then, the relative positions of the corresponding feature pairs were used to estimate the relative orientations of the robot at the time instants when the particular images were captured.

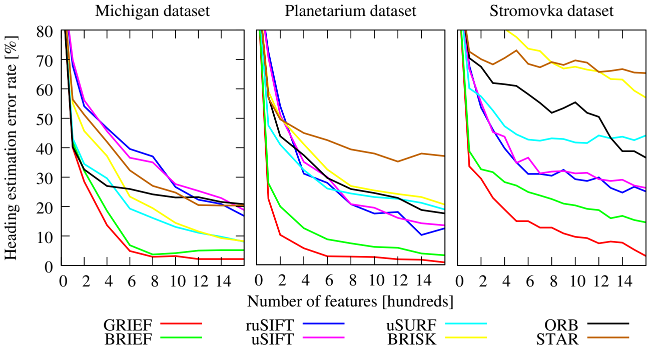


Fig. 8: Heading estimation error rates for various feature extractor types and different numbers of features extracted.

The Michigan and Planetarium datasets contain 5 different

[1] The ratio threshold value was set to 0.9, as this value showed the best performance for the feature extractors used in our evaluation.

locations with 12 images per location, which allows for $12 \times 11 \times 5 = 660$ comparisons for each feature type and threshold setting. The evaluation of the Stromovka dataset is based on 1300 (winter/summer) image pairs.

Figure 8 shows the dependence of the error rate on the number of extracted features and feature extractor type. Note that uSIFT and uSURF are 'upright' variants of SIFT and SURF, ruSIFT is the upright-root-SIFT and STAR means combination of STAR/SURF. We also established the average time required for feature extraction and matching on an Intel i5 PC running at 2.5 GHz. The results are depicted in Table I. Note that the OpenCV SIFT is faster than OpenCV SURF, which is a surprising feature of the OpenCV library used.

TABLE I: Error rates and computational times for 1600 extracted features

Extractor	GRIEF		ruSIFT		ORB		STAR	
		BRIEF		uSIFT		uSURF		BRISK
Planetarium [%]	1	3	12	13	17	19	38	21
Michigan [%]	2	5	17	19	21	9	20	8
Stromovka [%]	3	15	25	26	37	44	65	57
Time [ms/image]	37	37	169	170	232	42	39	95

VI. Conclusion

We report our results on the evaluation of image feature extractors to mid- and long-term environment changes caused by variable illumination and seasonal factors. The datasets used for evaluation capture appearance changes of two different outdoor environments throughout one year. The evaluation was taken from the navigational point of view - it was based on the feature extractor's ability to correctly establish the robot orientation and hence, keep it on the intended trajectory.

We noted that the upright-root-SIFT algorithm was outperformed by the BRIEF feature extractor that is based on bitwise comparisons of the pixel intensities around the detected keypoint. To further elaborate on this result, we have used an evolutionary algorithm to refine the comparison sequences that constitute the core of the BRIEF descriptor. Despite the fact that this adaptation was performed using only 12 images from a single location of one dataset, the improved feature, which we call GRIEF, outperforms its predecessor on other datasets as well. Overall, the image registration using upright-root-SIFT failed for 20% of the image pairs, whereas the GRIEF-based registration failed in only 3% of cases. In addition, the GRIEF feature is less computationally demanding than SIFT and thus it seems to be the most suitable feature descriptor for visual-based navigation systems operating in outdoor environments for long periods of time.

We hope that this evaluation will be useful for other researchers concerned with long-term autonomy of mobile robots in challenging environments and will help them to choose the most appropriate image feature extractor for their navigation and localization systems.

References

[1] J. Li and N. Allinson, "A comprehensive review of current local features for computer vision," *Neurocomputing*, 2008.

[2] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[3] D. Mukherjee, Q. JonathanWu, and G. Wang, "A comparative experimental study of image feature detectors and descriptors," *Machine Vision and Applications*, 2015.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International journal of computer vision*, vol. 94, no. 3, pp. 335–360, 2011.

[6] A. Gil, O. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual slam," *Machine Vision and Applications*, pp. 905–920, 2010.

[7] F. Dayoub and T. Duckett, "An adaptive appearance-based map for long-term topological localization of mobile robots," in *IROS*, 2008.

[8] M. Milford and W. Fraser, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *ICRA*, 2012.

[9] W. Churchill and P. Newman, "Practice makes perfect? managing and leveraging visual experiences for lifelong navigation," in *ICRA*, 2012.

[10] P. Neubert, N. Sünderhauf, and P. Protzel, "Appearance change prediction for long-term navigation across seasons," in *ECMR*, 2013.

[11] T. Krajník, J. P. Fentanes, O. M. Mozos, T. Duckett, J. Ekekrantz, and M. Hanheide, "Long-term topological localization for service robots in dynamic environments using spectral maps," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[12] C. McManus, B. Upcroft, and P. Newmann, "Scene signatures: localised and point-less features for localisation," in *Robotics: Science and Systems (RSS)*, 2014.

[13] N. Carlevaris-Bianco and R. M. Eustice, "Learning visual feature descriptors for dynamic lighting conditions," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2014.

[14] T. Krajník, S. Pedre, and L. Přeučil, "Monocular Navigation System for Long-Term Autonomy," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*. Montevideo: IEEE, 2013.

[15] C. Valgren and A. J. Lilienthal, "SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 157–165, February 28 2010.

[16] Z. Chen and S. T. Birchfield, "Qualitative vision-based path following," *IEEE Transactions on Robotics and Automation*, 2009.

[17] E. Royer et al., "Monocular vision for mobile robot localization and autonomous navigation," *Int. Journal of Computer Vision*, 2007.

[18] T. Krajník, J. Faigl, V. Vonásek et al., "Simple, yet Stable Bearing-only Navigation," *Journal of Field Robotics*, 2010.

[19] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[20] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, 2008.

[22] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: Center surround extremas for realtime feature detection and matching," in *European Conference on Computer Vision (ECCV)*. Springer, 2008.

[23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proceedings of the ICCV*, 2010.

[24] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, 2006.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *International Conference on Computer Vision*, Barcelona, 11/2011 2011.

[26] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceedings of the ICCV*, 2011.

[27] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *European Conference on Computer Vision*, 2010.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, 1981.

[29] "Stromovka dataset," [Cit: 2013-03-25]. [Online]. Available: http://purl.org/robotics/stromovka_dataset