Institute of Parallel and Distributed Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3292

Efficient matching of robust features for embedded SLAM

Zhen Peng

Course of Study:	Computer Science
Examiner:	PD Dr. Viktor Avrutin
Supervisor:	Torfi Thorhallsson PD Dr. Viktor Avrutin,
Commenced:	17. 01 2012
Completed:	12. 07 2012
CR-Classification:	I.2.10, I.4.7

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

In the day life, humans perceive various information from the environment, more than 80% of which are obtained through the visual system. All advanced animals have well-developed visual system to find the food source or identify the target in the environment. For us humans this is an innate ability. With the development of the computer technology, people try to use the camera to obtain the visual information from environment and convert it into the digital signals. The whole process of acquiring, processing, analyzing, and understanding of visual information by computer system is then developed as a new research field - *Computer Vision*. More and more mobile applications are equipped with camera for vision perception, environment analysis, decision making and localization. The motion of the camera system can be estimated by comparing the current frame with the previous frame. The process of finding the same objects on both images is called *Image Matching*. The image pair is called query image and reference image. Besides the set of matched points, a matrix describing the transformation between the two images is determined and used in the matching process. A simple case is that of a pure translation in the image. In other cases a more complex model is required to appropriately constrain the motion of features between the two views. This can include rotation, scaling, affine and perspective transformations of the image.

In general the image matching algorithms can be divided into the two categories: *Area Based Matching* (ABM), and *Feature Based Matching*(FBM) [GBG10]. The prominent difference is, ABM uses windows composed of intensity as the matching primitives in the matching step, FBN uses features extracted from image instead.

Direct use of the original intensity values makes full usage of the image information to distinguish different objects precisely. Processing of a large amount of information increases the computational complexity. Another Shortcoming of ABM is the sensitivity to the subtle differences between the two images, small intensity changes (for instance under different illumination conditions) have influence on the matching results. Because of the poor noise-resistibility, such ABM algorithms are usually only used for the precise matching problem between the two images without huge differences. In FBM the matching primitive is feature, such like points, lines, regions or global feature called structures, which usually composed of points, lines and regions. Among these feature types, points are the most used features. The main idea of FBM is detecting distinctive and robust features from images, based on the similarity of features to find the best match between the query image and reference image. Compared to the ABM, the required number of pixels for feature computation is significantly reduced. The FBM algorithms are less sensitive to the noise, the matching results rely on the

detected image features. This thesis mainly discusses such feature-based image matching methods. The figure 1.1 shows an example of image matching result.

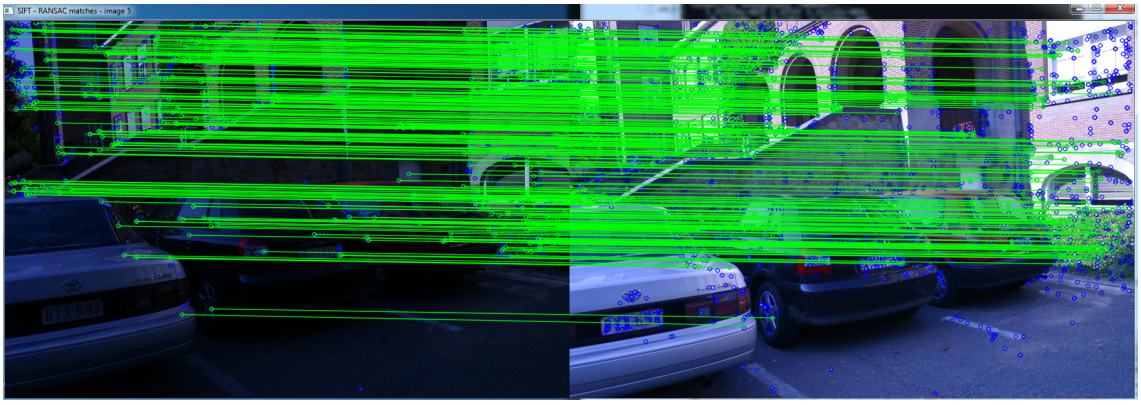


Figure 1.1.: Example of image matching based on features.

Because of the high efficiency, robustness and noise-resistibility, image matching based on the local point features has become a widely accepted and utilized method in the recent past, a wide range of feature detectors and feature descriptors have been proposed, the performance comparison between the most used descriptors is the purpose of this thesis.

1.2. Related work

The SIFT feature [Low99, Low04] is one of the most popular point features with outstanding performance. It has been proved that the SIFT algorithms can accurately find the matched feature points even under some extreme conditions [MS05]. But SIFT has an obvious drawback in the large amount of computation, which leads to long processing time. In a mobile application system, the performance limitation of embedded microprocessor must be considered. Later several variants of SIFT have been developed to optimize the steps of SIFT, such as PCA-SIFT [KS04] and Harris-SIFT [AAD09]. The SURF [BETG08] detector builds upon the SIFT but uses box filters to approximate the Gaussian in SIFT, has a faster computation speed compared to SIFT, it is still not fast enough for the requirements of the real-time application. More recently some faster algorithms have been introduced. Unlike SIFT and SURF, they use binary description instead of scalar valued vector description. Matching of binary strings is obviously more efficient than matching of vectors. Some of the new approaches sacrifice part of the performance to reach the advantages of short computing time. For instance, BRIEF[CLSF10] and AGAST [MHB$^+$10] are unable to find the correct matches when the scale is changed. Based on them, ORB [RRKB11] and BRISK[LCS11] are developed, they simultaneously maintain high performance and short computation time. In most of the above mentioned papers, the authors compared performance of the proposed approach with SIFT/SURF as standard. In order to evaluate the performance among different feature descriptors under varying situations, a comparison becomes necessary. In the previous

time, there are already several comparison studies. For instance, Mikolajczyk and Schmid [MS05] evaluated a variety of local descriptors including steerable filters [FA91], complex filters [SZ02], differential invariants [KD87], moment invariants [GMU96] and SIFT, and identified the SIFT algorithms as being the most resistant to common image deformations; Juan and Gwun [LJ09] compared SIFT, PCA-SIFT and SURF for scale changes, rotation, blur, illumination changes and affine transformation; Schmidt Kraft and Kasinski [SKK10] presented an evaluation of image feature detectors and descriptors for robot navigation. The new algorithms published in past two years like ORB and BRISK have not been mentioned and compared in the published papers. This thesis compares the new feature descriptors with the old classic approaches.

1.3. Disposition

The thesis is divided into seven chapters, each chapter is organized as follows. The next two chapters are intended to provide the reader with a solid background; they introduce first the basic knowledge of visual odometry and then review feature detection and matching methods where the most relevant algorithms are discussed in details. In Chapters 4 and 5 follow the evaluation and comparison of 5 new feature descriptors, which are published after year 2010, with two strong, widely used classic descriptors. Chapter 6 then presents the experimental evaluation on real-time video sequence. Chapter 7 concludes with a discussion of the impact of this work, comments on its limitations, and highlights future research directions.

2. Visual odometry

Visual odometry(VO) is the process of estimating the egomotion of an agent (e.g., vehicle, human, and robot) by analyzing the associated image from attached single or multiple cameras. Nister, Naroditsky and Bergen coined the term VO in their paper [NNB04] in 2004. Similar to wheel odometry, VO estimates the pose of the agent incrementally through examination of the changes that motion induces on the images of its onboard cameras. Under the condition of sufficient illumination in the environment and static scene with enough texture, VO works effectively. Another obviously advantage of VO is that the visual system is not effected by uneven terrain. It has been demonstrated that compared to wheel odometry, VO provides more accurate trajectory estimates, the relative position error is ranging from 0.1 to 2% [SF11]. VO has been used in a wide variety of robotic applications, such as on the Mars Exploration Rovers [MCM07, CG08]. Actually most of the early research in VO [Mor80, MS90, Mat89, SLC99, OMSM00] was done for planetary rovers and was motivated by the NASA Mars exploration program. The researchers were trying to develop all-terrain rovers with the capability to measure their 6-degree-of-freedom (DoF) motion in planetary environments, like uneven and rough terrains or other adverse conditions for traditional wheel odometry. This all-terrain capability makes VO an useful replacement or supplement to wheel odometry and other navigation systems such as *global positioning system* (GPS) [LWZ11], *inertial measurement units* (IMUs) [Kle08], and laser odometry (similar to VO, egomotion estimation by consecutive laser-scan-matching) [ABH+10, Ols09].

2.1. Pipeline

Most existing approaches to visual odometry are based on the following stages: (summarized in Figure 2.1)

1. Acquire input images: use either single cameras [SCS08, NNB04], stereo cameras [NNB04, CMR10], or omnidirectional cameras [SS08, Cor04] to capture the image sequence.
2. Image correction: apply image processing techniques for lens distortion removal, image enhancement, etc.
3. Feature detection: define interest operators, detect feature keypoints from the previous frame.

4. Feature matching or tracking: either matching features independently in both frames based on some similarity metrics or tracking the features extracted form previous frame in the current frame using a local search technique, such as correlation.
5. Motion estimation: compute the relative motion between the previous frame and the current frame. Depending on the dimensions of correspondences, there are three different approaches:
   a) 2D-to-2D: VO from image feature correspondences. The essential matrix for 2D image pairs is computed first, this requires at least five 2D-to-2D feature correspondences [Kru13, Nis03]. Then this essential matrix can be easily decomposed into rotation and translation[LH87]. After computation of relative scale, rescale the translation to obtain the complete transformation.
   b) 3D-to-3D: VO from 3D structure correspondences. Stereo images are required here. 3D features can be constructed by triangulation of matched features for each stereo pairs. The transformation is computed from 3D features.
   c) 3D-to-2D: VO from 3D structure and image feature correspondences. In the monocular case, 3D structure needs to be triangulated from two adjacent frame and then matched to 2D image feature in third frame. For the motion estimation, at least three frames are necessary in this case.

   It has been demonstrated by Nister et al. [NNB04] that 3D-to-2D motion estimation methods are more accurate than 3D-to-3D methods, because 3D-to-2D correspondences minimizes the image reprojection error instead of the 3D-to-3D feature position error.
6. Local optimization: After obtaining the motion information, the camera pose is computed by concatenation of relative motion with the previous pose. Finally, an iterative refinement (bundle adjustment) can be done over the last m frames to obtain a more accurate estimate of the local trajectory.

2.2. Feature tracking

As mentioned in the last section, there are two main approaches for finding feature points and their correspondences. An alternative to feature matching approach is feature tracking. Basically, feature tracking consistent of two steps: detecting a set of feature keypoints in the first frame only and then searching inside a suitable sized window in the subsequent frames for their corresponding matches, the position of the searching window correlates with the texture around the feature in the first image. The disadvantage of this approach is that features tend to drift. The feature tracking approach is more suitable when the images are captured by on-board cameras consequently at nearby location and the appearance deformation between image pairs is small. However, over long image sequences, the appearance of the features may change a lot, the texture in the subsequent frame may be rotated, scaled or skewed with respect to the texture in the first frame, the tracking becomes much more difficult. In this situation, feature matching performs better since its

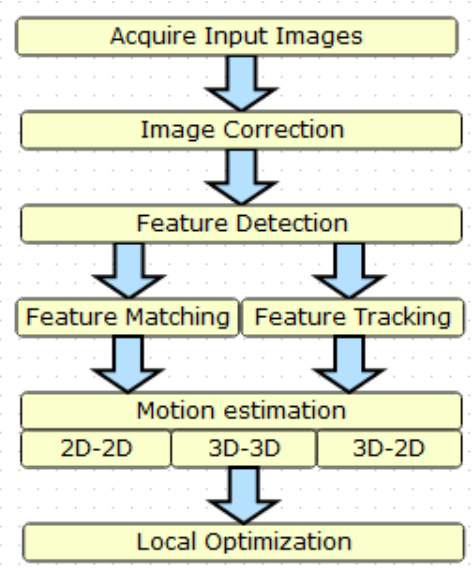


Figure 2.1.: A block diagram showing the pipeline of a visual odometry system.

features are extracted independently from images and matched based on similarity of their descriptors. Early research in VO concentrated on feature tracking approach because most experiments were conducted in the small-scale environments, where images were taken from nearby location. This situation has changed in the last decade, the focus has shifted to large-scale environments, and so the images are taken far from each other, the feature matching approach became more suitable. In the next chapter the most popular feature matching methods are discussed in details.

3. Feature description and matching

An image matching process can be divided into three stages: detection, description and matching. During the feature detection stage, some interest operator is applied on the images to find distinctive keypoints, which are likely to match well in other images. For point feature detector, such keypoints are corners or blobs. Corner is defined as a intersection of two or more edges. A blob is an image pattern with an intensity, color, and texture different from its surrounding region. Comparing to blob detector, Corner detectors are less distinctive but run faster. During the feature description stage, the detected features are described based on the neighbor pixels around it. Basically there are two type of descriptors: vector descriptor or binary descriptor. Vector descriptor is a feature vector with n dimensions, for instance n=128 for SIFT features. It stores more information, but it is difficult to find the nearest match in high dimensional space. Binary vector is a n-bit binary String consisting of 0 and 1. It can be processed quite fast with efficient algorithms. During the matching stage, each query feature is matched to the most similar feature in the reference image based on their descriptors.

3.1. Feature detector and descriptor

Some of the best known feature detectors and descriptors are introduced in this section.

3.1.1. SIFT

SIFT (*Scale Invariant Feature Transform*) feature was propose by David Lowe in 1999 [Low99] and improved in 2004 [Low04]. SIFT consists of four major stages : scale-space extrema detection, keypoints localization, orientation assignment and keypoint descriptor.

A scale pyramid is constructed first, the upper and lower scales of the image $I(x, y)$ are convolved with a *difference-of-Gaussian* (DoG) operator, which defined as:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

where $*$ is the convolution operation in x and y, σ presents the current scale, k is the constant multiplicative factor in scale space, and Gaussian

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

The local minima or maxima in scale space are taken as the potential feature points. Next step is rejecting unstable extrema with low contrast and the points which are poorly localized along an edge. All keypoints left are defined as SIFT features. Before describing the features, the image is smoothed by Gaussian. For each Gaussian smoothed image sample $L(x, y)$ at one scale, the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ is precomputed using pixel differences:

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \\ \theta(x,y) &= \tan^{-1}(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}) \end{aligned}$$

Based on $\theta(x, y)$ 36-bin orientation histogram is formed by taking values from the sample points around the keypoint. 36 bins cover the 360 orientation degree. Each sample added to the histogram is weighted by precomputed $m(x, y)$ and by a Gaussian around the keypoint. This Gaussian circular window avoids sudden changes in the descriptor with small changes in the position of the window. The main orientation of the keypoint is computed based on dominant directions of local gradients. The distribution of local gradients around the keypoint are summarized from 16x16 sample array into a 4x4 descriptor. The process is illustrated in Figure 3.1. This vector is then normalized to make SIFT descriptor more robust to illumination changes. Because the histogram is computed at the same scale as keypoint and the gradients are all rotated according to the main orientation of keypoint, SIFT descriptor are scale- and rotation invariant. Because of the outstanding invariance, SIFT descriptor is widely used in image matching process, although the 128 dimension of the descriptor vector reduce its real - time performance. This algorithm has its own patent, the patent holder is the University of British Columbia.

3.1.2. PCA-SIFT

PCA (*Principle component analysis*) is a standard technique for dimension reduction. Ke and Sukthankar combined the SIFT descriptor with PCA algorithm to reduce the dimension of SIFT descriptor vector [KS04].The PCA-SIFT has the same input as the standard SIFT descriptor: the sub-pixel location, scale, and dominant orientations of the keypoints. An eigenspace is computed to express the gradient images of local patches. The gradient image vector is then project into a compact feature vector. It is empirically determined that n = 20 is a good value for the dimensionality of the feature space[KS04], which results to significant space benifits. Comparing to SIFT, PCA-SIFT requires less storage, fewer components results a faster process time.

3.1.3. Harris-SIFT

Harris-SIFT is another variant of SIFT proposed by Azad, Asfour and Dillmann [AAD09]. They combined Harris interest points and the SIFT descriptor. The fast Harris corner detector

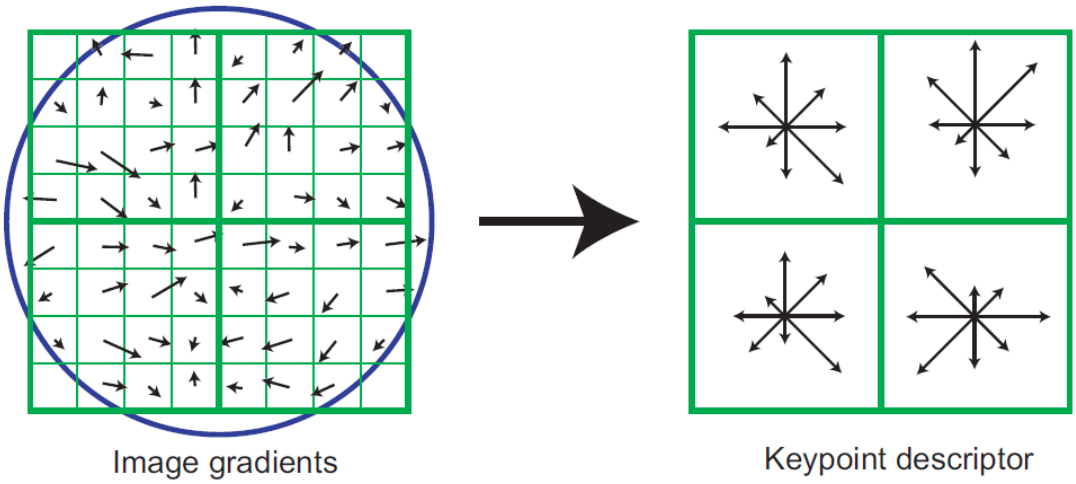


Figure 3.1.: Example of computing a SIFT descriptor. Local gradients around the keypoint are weighted by a Gaussian circle window (left) and summarized in 4 histograms with 8 orientation bins each (right). The standard SIFT descriptor uses a 16x16 sample array and 4x4 histograms, resulting in a 128-dimensional vector. Illustration taken from[Low04].

is used as the feature detector to replace the complex computation for feature detection in SIFT. Unlike SIFT there is no scale pyramid constructed. In order to retain the invariance to scale, three lower scales of the image are used for producing the SIFT descriptors. Since the most time-consuming part of standard SIFT is replaced, Harris-SIFT shows significant advantage of process time but less invariance comparing to SIFT.

3.1.4. SURF

The *Speeded-Up Robust Features* (SURF) method, proposed by Bay et al. [BETG08] essentially can be seen as an approximation to SIFT. SURF detector builds upon the SIFT but employ slightly different ways of detecting features. SIFT constructs scale pyramid, convolving the upper and lower scales of the image with a difference-of-Gaussian (DoG) operator and searching the local extreme in scale space. SURF scales filters up instead of iteratively reducing the image size. This avoids aliasing but limits scale invariance. A second substantially different between SURF and SIFT is that SURF uses 9x9 box filter to approximate the second-order Gaussian partial derivatives in SIFT.

During the description stage Haar wavelets is used, which allows to determine the gradient values in x and y direction. In order to obtain the dominant orientation, the Haar wavelet responses are computed for all points within the radius $6s$ of the detected feature point, where s is the scale at which this feature point was detected. Once the wavelet responses are calculated and weighted with a Gaussian, the dominant orientation is estimated by summing

of all horizontal and vertical responses within a sliding orientation window covering an angle of $\frac{\pi}{3}$. The Orientation with the longest vector is selected as the dominant orientation of the descriptor. A square window around the feature point with the size of $20s$ is divided into 4x4 square subregions and each subregion is divided into 5x5 regularly spaced sample points. Haar wavelet response for horizontal directions dx and vertical direction dy are computed at each sample points, then summed up over each subregion. The descriptor for each of the subregion consists of responses ant their absolute values of each principal directions:

$$v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$$

Therefore, the complete descriptor vector for all 4x4 sub-region has the length of 64. For reasonably fast processing time and robustness to typical image transformations, SURF became the de facto standard. SURF was patented by ETH Zurich, and the rights sold to Toyota.

3.1.5. FAST

FAST stands for *Features from Accelerated Segment Test* [RD06]. This corner detector consists of two steps. At first, a segment test is applied on each corner candidate P. Sixteen pixels around P are considered in this segment test. Let I_p denote the brightness of P and t a configurable threshold value, if n contiguous pixels in the circle are all brighter than $I_p + t$, or darker then $I_p - t$, the candidate P is defined as a corner. The Figure 3.2 illustrates the Twelve-point segment test corner detection in an image patch. It is demonstrated that the best results are obtained when n=9, the corresponding algorithm is called FAST-9 [RD06]. The ordering of questions, which neighbor pixel in the circle should be tested next, is learned by using the ID3 algorithm. As the segments test produces many adjacent responses around the interest point, non maximal suppression with a score function V, which is defined as:

$$V = \max \left(\sum_{x \subset S_{bright}} |I_{p \to x} - I_p| - t, \sum_{x \subset S_{dark}} |I_p - I_{p \to x}| - t \right)$$

is applied to remove corners which have an adjacent corner with higher V. This allows for precise feature localization. As the non maximal suppression is only performed to a small subset of image points, which passed the first segment test, the processing time remains short. While being efficient, FAST has proven to be reliable due to high repeatability [RPD08] and becomes one of the most popular feature detector being used in some real-time Applications such like Klein's PTAM [KM07] and Taylor's robust feature matching in 2.3 μs [TRD09].

3.1.6. AGAST

Similar to FAST, a approach called AGAST (*Adaptive and Generic Corner Detection Based on the Accelerated Segment Test*), which proposed by Mair et al. [MHB+10], is also based on the

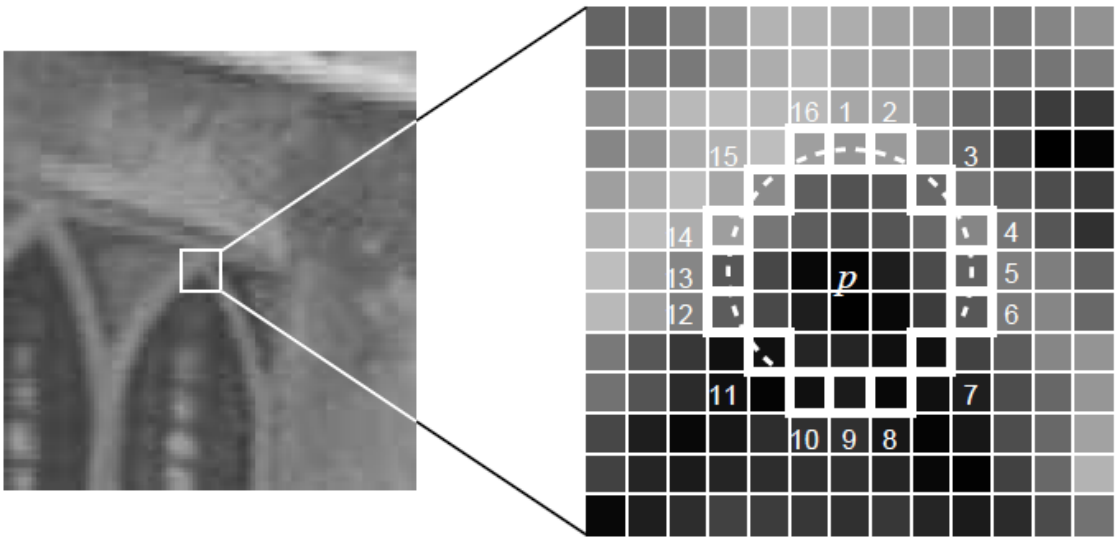


Figure 3.2.: Twelve-point segment test corner detection in FAST. If twelve contiguous pixels in the circle are all brighter than $I_p + t$, or darker than $I_p - t$, the candidate P is defined as a corner. Illustration taken from[RD06].

accelerated segment test . AGAST uses binary decision trees to complete the accelerated segment test. Two trees are constructed, one for homogeneous and one for structured regions. By combining two trees, the corner detector adapts to the environments automatically and provides the most efficient decision tree for the image region. AGAST does not have to be trained while preserving the same corner response and repeatability as the FAST corner detector.

3.1.7. BRIEF

The *Binary Robust Independent Elementary Features* (BRIEF) is a new feature descriptor proposed by Calonder [Cal10]. The essential different between BRIEF and previously mentioned descriptor is that BRIEF describes the features with binary string instead of vector. After feature detection stage, the feature patch is smoothed to reduce the noise-sensitivity, thus increasing the stability and repeatability of the descriptors. In smoothed Patch, 128 pixel pairs around keypoint are selected for binary tests, which compares the intensity of both pixels, value 1 means the first value is bigger than the second, 0 otherwise. The location of such pixel pairs is selected randomly. After several experimental evaluations, it is shown that the sampling from an isotropic Gaussian distribution performs best: $(X, Y) \sim i.i.d. Gaussian(0, \frac{1}{25}S^2)$, where S is the size of the feature patch. The Figure 3.3 shows the location of pixel pairs with this sampling approach. After the binary tests on these pixel pairs, a 128 bit binary string is build as the BRIEF descriptor. The BRIEF descriptor is quite fast to construct and to match because of its binary nature, but is not rotation and scale invariant.

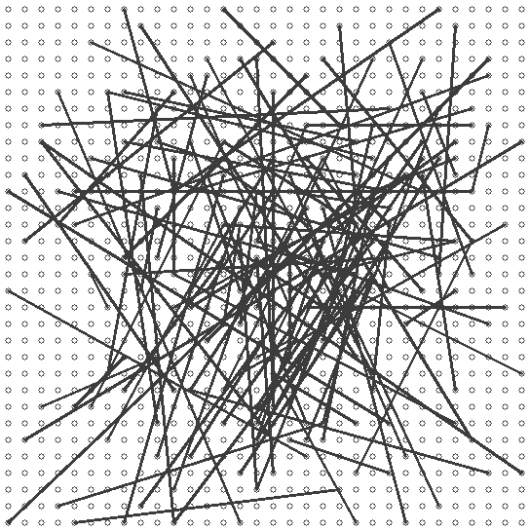


Figure 3.3.: Location of sampling pixel pairs in BRIEF. Illustration taken from[Cal10].

3.1.8. ORB

The ORB descriptor is developed by Rublee et al. [RRKB11] in 2011 based on BRIEF in order to cover the shortcoming of rotation and scale variance of BRIEF. ORB uses FAST-9 approach as the feature detector. After feature detection in a scale pyramid, all keypoints are sorted in a line based on the Harris corner measure, only top N points are picked. A metric called intensity centroid C is computed with the moments m of the patch:

$$\begin{aligned} m_{pq} &= \sum_{x,y} x^p y^q I(x,y) \\ C &= (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \end{aligned}$$

Constructing a vector from the center of the patch to the centroid, the orientation of the patch then simply is:

$$\theta = \arctan 2(m_{01}, m_{10})$$

Using the pre-computed patch orientation θ and the corresponding rotation matrix R_θ to rotate the feature patch, then BRIEF descriptor is applied on rotated features and records the binary string as ORB descriptor. ORB descriptor can be quite fast processed because of the binary nature and keeps the rotation and scale invariant at the same time.

3.1.9. BRISK

The *Binary Robust Invariant Scalable Keypoints* (BRIEF) is another newly published feature descriptor proposed by Leutenegger [LCS11]. An image pyramid is constructed while the inter-octaves between the scale layer are also considered. During the feature detection stage, the local extrema in the scale space based on FAST score are searched for. The FAST score is defined as the maximum threshold still considering an image point a corner in FAST 9 detector. During the feature description step, BRISK samples the neighborhood of the keypoints in a circle patter showed in figure 3.4. All sampling-point pairs are divided in to subset of short-distance pairings and another subset of long-distance parings, based on the distance between the two sampling-points. The long-distance parings are used for the local gradients computation to obtain the orientation of the BRISK descriptor and the short-distance pairings are used for building the BRISK descriptor with binary test. The use of scale pyramid and rotation with orientation makes BRISK descriptor scale and rotation invariant, and meanwhile the binary descriptor obtain the advantage on processing speed.

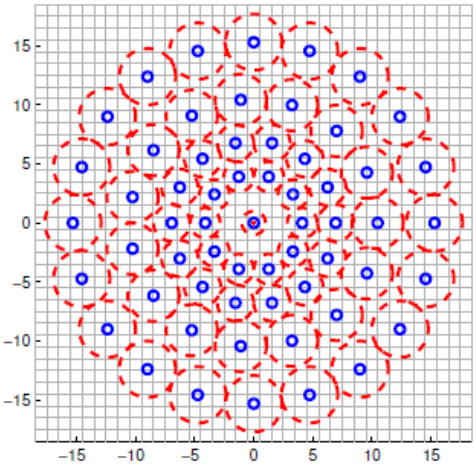


Figure 3.4.: Sampling patter of BRISK descriptor. Illustration taken from[LCS11].

3.2. Descriptor matching

Based on the form of descriptor, the descriptor matching process can be divided into the two categories: vector descriptor matching and binary descriptor matching. Among all in last section mentioned feature descriptors, SIFT, PCA-SIFT, Harris-SIFT and SURF are vector descriptors, while BRIEF, ORB and BRISK are binary descriptors. For a vector descriptor, such as SIFT, which has a 128-dimensional feature vector, there are no known algorithms that can identify the nearest neighbor of points in high dimensional spaces that are any more efficient than exhaustive search [Low04]. Even the best algorithms, such as the k-d tree [FBF77] has been proved having no speedup over exhaustive search for more than

about 10 dimensional spaces. Usually an approximate algorithm called *Best-Bin-First* (BBF) algorithm [BL97] is used to obtain the closest neighbor with high probability. Compared to vector descriptor matching, binary descriptor matching is much easier. The closest match are the descriptors with the smallest Hamming distance. The number of bits different in the two descriptors measures their dissimilarity. Notice that the Hamming distance calculation can be reduced to a bitwise XOR followed by a bit count, which can both be computed quite efficiently on today's computer architectures. Matching on binary descriptors requires significantly less time than vector descriptors.

3.3. Optimization

3.3.1. Cross check filter

Nearest neighbor matching methods will always return a match, even if the feature is not present in the reference image. This inevitably leads to a number of false matches. A cross check filter is applied at feature matching process to increase the accuracy of the feature matching result by double matching the image in both directions. The steps of the cross check approach summarized as follows:

1. For each query feature find one or more the most matching features in reference image.
2. Switch the reference image and query image.
3. For each feature from reference image, find one or more the most matching features in query image.
4. Only the feature pairs, which are matching features to each other, are accepted as matches.

Although the double matching increases the processing time, cross check filter can remove some unreliably matches before the results are returned. In the next three chapters, an experiment evaluation for the performance of cross check filter is conducted on static image pairs and also in a real-time application.

3.3.2. RANSAC

The set of matched points usually contains a number of false matches or outliers. Possible causes of outlier are image noise, occlusions, blur and too complex image deformation, which the system does not account to the corresponding complex mathematical model. Outlier removal is important for the accuracy of the motion estamation in a VO system. The *random sample consensus* (RANSAC) [FB81] is the most used approach to remove the outliers for model estimation. The basic idea of RANSAC is to compute model hypotheses from randomly-sampled small sets of data points and then verify these hypotheses on the other

data points. The hypothesis that shows the highest consensus with the other data is selected as solution [NNB04]. The steps of RANSAC is summarized as following:

1. Randomly select a small subset S from the whole dataset D.
2. Calculate the most likely model based on the data points from S.
3. Apply this model on other points from D, compute the distance for each point.
4. Construct the inlier set based on the distance computed from step 3.
5. Store the inlier set with the largest consensus so far.
6. Repeat the step 1 to select a new test subset, until the maximum number of iterations is reached.
7. The subset with the maximum number of inliers is chosen as the solution.
8. Estimate the model using all inliers in best subset from step 7.

The RANCAC algorithm is easy to implement and removes the outliers efficiently for model estimation. The all experiments conducted in the next three chapters use RANSAC for outlier removel.

4. Descriptor comparison

As stated in section 1.2, the main goal of this thesis is to compare the performance of all popular feature descriptors. The following six feature descriptors were chosen for this comparison: SIFT, SURF, BRIEF, ORB, BRISK and SU-BRISK (a variant of BRISK). The feature descriptors in this list, with the exception of SIFT and SURF, were all proposed within the last two years, and are therefore quite new. As any VO system can be applied in both indoor and outdoor environment, general deformations, such as illumination change, rotation and scale change and view change are considered in this thesis. In the following section 4.1 the implementation of the whole comparison program are introduced. The next section 4.2 reviews the test image sequences which were used for the comparison. Section 4.4 defines the performance metrics. And the final section 4.5 presents the comparison results on these test image sequences using the above metrics.

4.1. Implementation

OpenCV is a free open-source library intended for use in image processing, computer vision and machine learning areas. It provides a huge amount of image matching algorithms. This library is well developed, all detector, descriptor and matcher classes have uniform interfaces. This class structure brings advantages for the implementation of a comparison program. The following 4 feature descriptors are available In OpenCV: SIFT, SURF, BRIEF and ORB. Since BRIEF is a descriptor without feature detection, FAST is , FAST is added for the feature detection. The author of AGAST and BRISK provide also a public implementation based on OpenCV interfaces. Beside the original BRISK feature, this implementation includes 3 variants, first one called U-BRISK, which is not rotation invariant but scale invariant; SU-BRISK, which neither rotation invariant nor scale invariant and the last one called S-BRISK, which is rotation invariant but not scale invariant. SU-BRISK is chosen from this list and included into the comparison program as a candidate, AGAST is used as feature detector for SU-BRISK. The whole comparison program is developed in C++. To retain the compatibility of different platforms, CMake is used as build system. An installation introduction is appended as an attachment in the end of this thesis (see Appendix A).

4.2. Datasets

Eight publicly available test image sequences are used to compare the methods on real-world data[1]. They are designed to test robustness to typical image disturbances that occur in real-world scenarios. As shown in figure 4.1, they include
illumination changes: *Light*,
image blur: *Trees*, *Bikes*
rotation and zoom : *Bark*, *Boat*,
viewpoint changes: *Graffiti*, *Wall*,
and compression artifacts: *Jpg*.

Figure 4.1.: Part of test images, showing the first and the last image in each sequence used for comparison purpose. A are the illumination changed images, B and C are the blurred images, D and E are the rotation and scale changed images, F and G are the affine transformed images, H are the compressed Jpg-images.

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/

There are six images in each sequence, by matching the first image to the remaining five, each sequence contains five image pairs. The corresponding five homography matrices, which describe the projective transformation between the image pair, are also given. With them the ground truth can be computed. Note that the given homography describes the transformation from the first image to the other images. In the real image matching process in comparison program, the first image is used as the reference image, the transformation information from the query image to reference image is required, therefore, the homography has to be inverted first. The five pairs in each sequence are sorted in order of increasing changes so that pair 1-6 is much harder to match than pair 1-2. Another reason for choosing this dataset is that many published papers relevant to image matching have also used these images, such like in [LJ09, KS04, LCS11, CLSF10, Cal10]. Using the same datasets makes the conclusion in this thesis comparable with other papers.

4.3. Drawing Configuration

It is possible to draw the matching results as shown in figure 1.1. In the source code main.cpp there are six configuration variables. The first **DRAW_IMAGE_MODE** enables this drawing function. The rest of them control what kind of information are displayed on screen:

- **DRAW_MATCHES** shows the matching pairs after description matching.
- **DRAW_STANDARD_MATCHES** shows the ground truth obtained by given homography.
- **DRAW_RANSAC_MATCHES** shows the matching result after RANSAC.
- **DRAW_RICH_KEYPOINTS_MODE** shows a circle around keypoint with keypoint size and orientation.
- **DRAW_OUTLIERS_MODE** shows all outliers.

The default configuration is shown in Figure 4.2.

```
34
35  #define DRAW_IMAGE_MODE             0
36  #define DRAW_MATCHES                1
37  #define DRAW_STANDARD_MATCHES       1
38  #define DRAW_RANSAC_MATCHES         1
39
40  #define DRAW_RICH_KEYPOINTS_MODE    1
41  #define DRAW_OUTLIERS_MODE          0
42
```

Figure 4.2.: Default drawing configuration in Descriptor Comparison program.

4.4. Performance Metrics

There are 7 different metrics used to evaluate the performance of feature descriptors from all aspects.

4.4.1. Keypoints

The number of feature points (also called keypoints) detected and matched between the image pair is one of the most important and intuitive measure for comparison the performance of all the descriptors. The whole process contains the following steps:

1. Apply feature detector on the reference image (the first image in each sequence), the number of detected features is recored as N_{ref}.
2. Apply feature detector on the query image (one of the remaining five images), the number of detected features is recored as N_{qry}. Because all the images in the sequence are sorted in order of increasing changes, the first image has always the best quality, so that more feature points can be detected in the reference image then in the query image, in other words $N_{ref} > N_{qry}$.
3. With help of the given homograhy, each keypoint in the query image is checked, if there is a matched keypoints in the reference image, such matches called correspondence. The number of the correspondence N_{csp} represents the theoretical maximal number of matches, which can be found after image matching.
4. Apply feature descriptor on both images, now all the features are described as a vector or a binary string.
5. Apply descriptor matcher to find the matching keypoints based on description.
6. Apply RANSAC to refine the matching results. With help of the given homograhy, the correct, false and missing matches can now be classified. Figure 4.3 illustrates the relation between matches.

4.4.2. Repeatability

Repeatability represents the ability to detect the same point in the scene under viewpoint and lighting changes and subject to noise[Cal10]. The value of *Repeatability* is calculated as:

$$Repeatability = \frac{|correspondences|}{|query\ keypoints|}$$

Repeatability is only relevant to the feature detector, nothing about feature descriptor or descriptor matcher. The higher value of *Repeatability* , the better performance of feature detector.

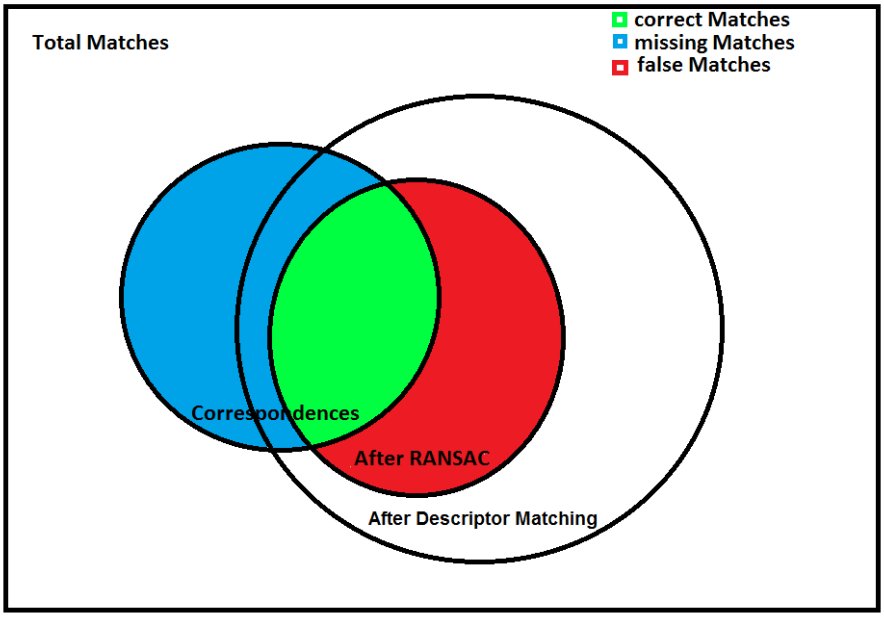


Figure 4.3.: Relation between correct, false and missing matches.

4.4.3. Recall

Recall represents the ability to find the correct matches based on the description of detected features, The value of *Recall* is calculated as:

$$Recall = \frac{|correct\ matches|}{|correspondences|}$$

Because the detected features are already determined, recall only shows the performance of the feature descriptor and descriptor matcher. The higher value of *Recall* , the better performance of descriptor and matcher.

4.4.4. Efficiency

The metric *Efficiency* combines the *Repeatability* and *Recall* . It is defined as:

$$Efficiency = Repeatability * Recall = \frac{|correct\ matches|}{|query\ keypoints|}$$

Efficiency measures the ability of the whole image matching process, it is relevant to all three steps: detection, description and matching. The higher value of *Efficiency* , the more accurate the image matching.

The three metrics Repeatability, Recall and Efficiency are also called a quality measure.

4.4.5. Duration

Duration of the entire image process is another important measure. Note that the time beginning at the feature detection and ending after matching are recorded, the RANSAC step is not included.

4.4.6. Speed

The number of detected keypoints is quite different depending on different feature detection approach. For instance, for the first image pair in illumination change sequence, SIFT detects 1770 keypoints, FAST (detector for BRIEF) 7634, ORB only 702 (adjustable limitation). Using single metric *duration* to compare the descriptors in time domain is not fair, therefor a new metric *Speed* is added as:

$$Speed = \frac{Duration}{|query\ keypoints| + |reference\ keypoints|}$$

It defines the average processing time for one feature.

4.4.7. Average Distance

With help of the given homography matrices, the position error of the matches can be computed. In the following experiments the distance is measured in pixels. Only the correct matches are considered here. *Average Distance* is defined as:

$$Average\ Distance = \frac{total\ position\ error\ among\ correct\ matches}{|correct\ matches|}$$

The smaller *Average Distance* , the more accurate the matching results.

During the execution of the comparison program, all the output information and results displayed on the screen are also written into a file called 'log.txt' and saved into the file system. At the same time, 7 files for the above mentioned performance metrics are generated with corresponding name. These 7 files contain only the numbers. Then these data are input into a Microsoft Excel work sheet manually for analysis and visualization. The visualized results are shown as the final comparison results in the next section.

4.5. Results

4.5.1. Illumination change

Illumination change is one of the most common changes in the real life. Figure 4.4 shows the test image sequence. The images are sorted in the order of increasing illumination change,

the last image being much darker than the first image. Figure 1.1 shown in the first chapter as example is the matching result after applying RANSAC between the 1st and the 5th image using SIFT feature.

Figure 4.4.: Test image sequence for illumination changes - *Light* sequence.

Keypoints The following table 4.1 summarizes the number of detected feature keypoints in the query image, the number of correspondences and correct matches found after RANSAC. With decreasing illumination condition in the image, the number of detected feature points shows also a decreasing trend (except ORB, because the parameter in ORB is set to return the top 702 feature keypoints based on Harris corner measure). BRIEF (FAST as detector) detects the most feature points and matches, on the image pair 1-2 more than 7500 keypoints are detected and more then 6000 of them find the correct matches eventually. Even when matching to the darkest image pair 1|6, around 3000 feature are correctly matched. SURF and SU-BRISK detect second most feature keypoints, about 1/3 - 1/4 of BRISK. On the darkest 3 images, BRISK ORB detect the least feature points, only 300,however, more then 100 correct matches are found successfully.

Repeatability The following table 4.2 summarizes the value of *Repeatability*on this sequence. Except FAST (detector of BRIEF), the Repeatability of all other descriptors are decreased with the reduction of illumination. The Repeatability of FAST is slightly increased and remained at 0.9. AGAST (detector for SU-BRISK) and FAST perform best, even when matching to the darkest image pair 1|6, the Repeatability keeps around 0.9. Performance of SIFT and ORB is

Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
ILU-1-2	1770	1154	967	2403	2046	1714	7634	6855	6301
ILU-1-3	1571	992	785	1973	1629	1277	6439	5845	5351
ILU-1-4	1339	809	639	1576	1269	957	5356	4820	4432
ILU-1-5	1198	729	524	1216	940	612	4467	4070	3710
ILU-1-6	1022	567	375	1001	734	453	3601	3285	2984
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
ILU-1-2	702	527	407	1001	804	577	2167	1996	1938
ILU-1-3	702	448	332	798	609	429	1825	1668	1613
ILU-1-4	702	418	283	639	462	268	1409	1286	1255
ILU-1-5	702	387	236	523	360	201	1111	1022	970
ILU-1-6	702	362	184	330	215	110	822	742	698

Table 4.1.: Results of detected keypoints on *Light* sequence. (N_{qry}: number of detected query features, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC)

relatively poor, under the insufficient illumination condition, the value of Repeatability falls down under 0.6 rapidly.

Repeatability	SIFT	SURF	FAST	ORB	BRISK	SU-BRS
ILU-1-2	0.652	0.851	0.898	0.751	0.803	0.921
ILU-1-3	0.631	0.826	0.908	0.638	0.763	0.914
ILU-1-4	0.604	0.805	0.900	0.595	0.723	0.913
ILU-1-5	0.609	0.773	0.911	0.551	0.688	0.920
ILU-1-6	0.555	0.733	0.912	0.516	0.652	0.903

Table 4.2.: Results of *Repeatability* on *Light* sequence.

Recall The following table 4.3 summarizes the value of *Recall* . Basically the value of Recall decreases with increasing illumination change. BRIEF and SU-BRISK perform best, especially applying SU-BRISK on the first three image pairs obtain the high value 0.96. The result of other descriptors perform similar, the value of recall keeps between 0.5 and 0.75.

Efficiency The following table 4.4 summarizes the value of *Efficiency* . Efficiency combines the results of Repeatability and Recall. BRIEF and SU-BRISK show outstanding performance on Efficiency same as on both Repeatability and Recall metrics.SIFT, ORB and BRISK perform poorly in comparison, on the last three image pairs, the value of Efficiency is less then 0.5, the worst value is the 0.26 when applying ORB on the last image pair.

Recall	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
ILU-1-2	0.838	0.838	0.919	0.772	0.718	0.971
ILU-1-3	0.791	0.784	0.915	0.741	0.704	0.967
ILU-1-4	0.790	0.754	0.920	0.677	0.580	0.976
ILU-1-5	0.719	0.651	0.912	0.610	0.558	0.949
ILU-1-6	0.661	0.617	0.908	0.508	0.512	0.941

Table 4.3.: Results of *Recall* on *Light* sequence.

Efficiency	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
ILU-1-2	0.546	0.713	0.825	0.580	0.576	0.894
ILU-1-3	0.500	0.647	0.831	0.473	0.538	0.884
ILU-1-4	0.477	0.607	0.827	0.403	0.419	0.891
ILU-1-5	0.437	0.503	0.831	0.336	0.384	0.873
ILU-1-6	0.367	0.453	0.829	0.262	0.333	0.849

Table 4.4.: Results of *Efficiency* on *Light* sequence.

Duration The following table 4.5 summarizes the *Duration* results. Because of the computation complexity, SIFT needs significantly more time than any other descriptors. The advantage of binary description is fully demonstrated on this metric. ORB, BRISK and SU-BRISK run almost 50 times faster than SIFT. BRIEF processes a huge amount of feature keypoints, it does not show the superiority on the duration of whole image matching process.

Duration	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
ILU-1-2	6.006	2.274	2.035	0.126	0.151	0.690
ILU-1-3	5.619	2.097	1.732	0.122	0.136	0.595
ILU-1-4	5.387	1.919	1.45	0.127	0.121	0.468
ILU-1-5	5.173	1.709	1.24	0.121	0.113	0.371
ILU-1-6	4.932	1.611	1.01	0.118	0.097	0.286

Table 4.5.: Results of *Duration* on *Light* sequence. (unit: s)

Speed The following table 4.6 summarizes the result of *Speed*. After average duration through the number of keypoints, BRIEF shows its superiority as a binary descriptor. BRISK and SU-BRISK perform best, even better then BRIEF and ORB. The disadvantage of using SIFT and SURF is made apparent.

Average Distance The following table 4.7 summarizes the value of *Average Distance* . The accuracy of SIFT is obvious higher than any other descriptors, it contains the least position error. BRIEF and ORB perform relatively poor, but still in a acceptable range.

Speed	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
ILU-1-2	1.521	0.419	0.117	0.090	0.065	0.138
ILU-1-3	1.498	0.419	0.107	0.087	0.064	0.127
ILU-1-4	1.531	0.417	0.096	0.090	0.061	0.110
ILU-1-5	1.531	0.403	0.087	0.086	0.061	0.094
ILU-1-6	1.540	0.400	0.076	0.084	0.058	0.078

Table 4.6.: Results of *Speed* on *Light* sequence. (unit: ms)

avg. Distance	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
ILU-1-2	0.388	0.664	0.918	0.953	0.768	0.840
ILU-1-3	0.418	0.712	0.978	0.985	0.839	0.832
ILU-1-4	0.531	0.830	1.001	1.083	0.989	0.796
ILU-1-5	0.633	0.867	1.002	1.080	0.895	0.850
ILU-1-6	0.623	0.929	0.994	1.024	1.031	0.921

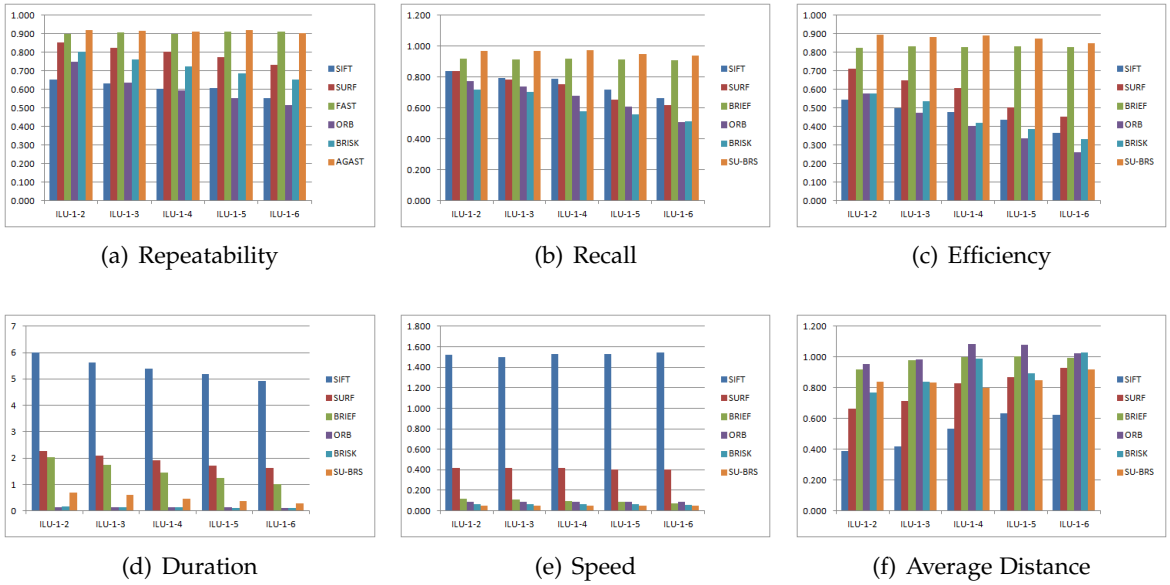
Table 4.7.: Results of *Average Distance* on *Light* sequence.

Summary Figure 4.5 shows the visualized comparison results. In conclusion, under the single illumination changes, BRIEF overall performances most prominent, obtains the highest value on Repeatability, Recall, and Efficiency metrics. Because there is no rotation and scale change in the image data, methods assuming zero rotation and scale change have an unfair advantage over invariant methods. It is therefore to be expected that ORB and BRISK perform less good then BRIEF and SU-BRISK. In time domain, the advantage of using binary description compare to vector description is quite obvious. Because of a large amount of detected keypoints, BRIEF lost the superiority on duration of whole image matching process. Despite of high accuracy, large time consumption for complex computation makes SIFT less competitive.

4.5.2. Blur

The two most likely causes of image blur are loss of focus and motion blur. Figure 4.6 shows the test image sequences. Comparing the two sequences, most objects in *Bikes* sequence have clear contour, while the *Trees* sequence is more unstructured.

Keypoints The following table 4.8 summarizes the number of detected feature keypoints in the query image, the number of correspondences and correct matches found after RANSAC. In *Bikes* sequence, BRIEF detects the most keypoints, while ORB, BRISK and SU-BRISK detect only small number of features. Particularly on the last image pair, BRISK and SU-BRISK find no correct matches at all. Compared to *Bikes* sequence, unstructured images from *Trees* sequence lead to a large amount of feature points. Except ORB with keypoints limitation, all other detectors detect more than 7000 features on the first two image pairs. FAST detects



(a) Repeatability (b) Recall (c) Efficiency

(d) Duration (e) Speed (f) Average Distance

Figure 4.5.: Comparison results on *Light* sequence.

even more than 35000 feature points and BRIEF matches more than 23000 successfully. On the last two difficult image pairs, BRIEF and SU-BRISK obtain sufficient number of correct matching, while other descriptors can hardly find any correct matches.

Repeatability The following table 4.9 summarizes the value of *Repeatability*. In the *Bikes* sequence, SURF performs best and is the most stable method, unlike FAST, which gets high Repeatability value on the first two image pairs and falls down rapidly when the blur effect is increased. On the last three image pairs, BRISK fails to produce any correct matches. In the *Trees* sequence, FAST performs best, even on the last image pair, retains the Recall value of 0.9. SIFT and ORB performance is relatively poor, particularly ORB gets a Recall value of 0.13 on the last image pair. Comparing the two sequences, SIFT performs similarly on both sequences, all other detector perform better on *Bikes* sequence than on the *Trees* sequence while for FAST and BRISK the difference in performance is opposite.

Recall The following table 4.10 summarizes the value of *Recall* . In the *Bikes* sequence, same as with the Repeatability measure, BRIEF performs best while BRISK performs worst. In *Trees* sequence, the results on the last two image pairs are quite bad, with the exception of BRIEF and SU-BRISK, almost no correct matches are found by other descriptors. Comparing the two sequence, all descriptor perform better on the *Bikes* sequence than on the *Trees* sequence.

Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
BL1-1-2	2053	1079	788	1977	1771	1573	2908	2677	2518
BL1-1-3	1335	799	563	1726	1544	1299	1877	1752	1588
BL1-1-4	735	467	316	1328	1160	920	967	868	708
BL1-1-5	510	320	235	1069	949	684	565	440	336
BL1-1-6	345	201	137	814	660	437	290	196	94
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
BL1-1-2	702	581	483	214	207	187	462	447	443
BL1-1-3	702	552	430	82	79	72	224	215	212
BL1-1-4	562	410	241	6	6	0	35	26	16
BL1-1-5	344	185	76	1	1	0	14	5	4
BL1-1-6	164	44	2	1	0	0	1	0	0
Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
BL2-1-2	9548	4928	1876	8209	4880	2113	38235	33971	23683
BL2-1-3	14769	6827	1900	7321	4135	1637	28516	25569	14405
BL2-1-4	10918	4779	448	6160	3083	727	16582	15051	6508
BL2-1-5	5410	2572	1	5659	2833	29	11716	10800	3515
BL2-1-6	3205	1493	0	4133	1971	0	7883	7154	1702
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
BL2-1-2	702	389	246	14476	10158	4976	20095	15577	11693
BL2-1-3	702	321	186	8987	6663	2710	12890	10384	6783
BL2-1-4	702	232	98	3362	2427	653	5373	4311	2217
BL2-1-5	702	195	0	1013	682	10	2205	1738	727
BL2-1-6	702	92	0	274	142	0	718	504	106

Table 4.8.: Results of detected keypoints on *Bikes* and *Trees* sequence. (N_{qry}: number of detected query Keypoints, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC)

(a) *Bikes* sequence.

(b) *Trees* sequence.

Figure 4.6.: Test image sequences for Blur - *Bikes* and *Trees* sequence.

Efficiency The following table 4.11 summarizes the value of *Efficiency*. The characteristics of Efficiency is basically similar to Recall, BRIEF performs best among the six descriptors.

Duration The following table 4.12 summarizes the result of *Duration*. In the *Bikes* sequence, ORB, BRISK and SU-BRISK finish the matching process in short time, while SIFT requires obviously more time. In THE *Trees* sequence, a huge amount of features are detected, the processing time for other descriptors is therefore significantly increased with the only

Repeatability	SIFT	SURF	FAST	ORB	BRISK	AGAST
BL1-1-2	0.526	0.896	0.921	0.828	0.967	0.968
BL1-1-3	0.599	0.895	0.933	0.786	0.963	0.960
BL1-1-4	0.635	0.873	0.898	0.730	1.000	0.743
BL1-1-5	0.627	0.888	0.779	0.538	1.000	0.357
BL1-1-6	0.583	0.811	0.676	0.268	0	0
BL2-1-2	0.516	0.635	0.998	0.554	0.854	0.925
BL2-1-3	0.605	0.565	0.897	0.457	0.741	0.806
BL2-1-4	0.438	0.500	0.908	0.330	0.722	0.802
BL2-1-5	0.475	0.501	0.922	0.278	0.673	0.788
BL2-1-6	0.466	0.477	0.908	0.131	0.518	0.702

Table 4.9.: Results of *Repeatability* on *Bikes* and *Trees* sequence.

Recall	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
BL1-1-2	0.730	0.888	0.941	0.831	0.903	0.991
BL1-1-3	0.705	0.841	0.906	0.779	0.911	0.986
BL1-1-4	0.677	0.793	0.816	0.588	0.000	0.615
BL1-1-5	0.734	0.721	0.764	0.411	0.000	0.800
BL1-1-6	0.682	0.662	0.480	0.045	nan	nan
BL2-1-2	0.381	0.433	0.697	0.632	0.490	0.751
BL2-1-3	0.278	0.396	0.563	0.579	0.407	0.653
BL2-1-4	0.094	0.236	0.432	0.422	0.269	0.514
BL2-1-5	0.000	0.010	0.325	0.000	0.015	0.418
BL2-1-6	0.000	0.000	0.238	0.000	0.000	0.210

Table 4.10.: Results of *Recall* on *Bikes* and *Trees* sequence.

Efficiency	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
BL1-1-2	0.384	0.796	0.866	0.688	0.874	0.959
BL1-1-3	0.422	0.753	0.846	0.613	0.878	0.946
BL1-1-4	0.430	0.693	0.732	0.429	0.000	0.457
BL1-1-5	0.461	0.640	0.595	0.221	0.000	0.286
BL1-1-6	0.397	0.537	0.324	0.012	0.000	0.000
BL2-1-2	0.196	0.275	0.696	0.350	0.418	0.694
BL2-1-3	0.168	0.224	0.505	0.265	0.302	0.526
BL2-1-4	0.041	0.118	0.392	0.140	0.194	0.413
BL2-1-5	0.000	0.005	0.300	0.000	0.010	0.330
BL2-1-6	0.000	0.000	0.216	0.000	0.000	0.148

Table 4.11.: Results of *Efficiency* on*Bikes* and *Trees* sequence.

exception of ORB. Particularly BRIEF takes more the 140 seconds to process the first image pair, ORB limits the number of features and keeps outstanding processing time.

Duration	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
BL1-1-2	8.020	2.633	3.331	0.195	0.100	0.162
BL1-1-3	6.753	2.394	2.240	0.165	0.065	0.101
BL1-1-4	5.753	2.000	1.225	0.197	0.047	0.055
BL1-1-5	5.674	1.847	0.741	0.135	0.047	0.046
BL1-1-6	5.359	1.706	0.478	0.118	0.041	0.045
BL2-1-2	42.257	10.979	143.501	0.294	43.432	34.617
BL2-1-3	58.394	10.229	118.542	0.282	22.687	22.677
BL2-1-4	47.801	8.915	62.971	0.265	8.554	9.596
BL2-1-5	28.245	8.321	44.511	0.247	3.006	4.279
BL2-1-6	21.297	6.838	29.701	0.229	0.976	1.501

Table 4.12.: Results of *Duration* on *Bikes* and *Trees* sequence. (unit: s)

Speed The following table 4.13 summarizes the results of *Speed* . Observing each sequence, the results is similar as in the *Light* sequence 4.5.1: BRISK and SU-BRISK run fastest while SIFT runs quite slowly. Comparing the two sequences, All descriptors take at least two times time to process one feature point on images from *Trees* sequence then *Bikes* sequence, with exception of ORB and SURF. This is because the large number of keypoints increases the time to matching exponentially. In summary, ORB performs best in this case.

Speed	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
BL1-1-2	1.536	0.538	0.254	0.139	0.070	0.065
BL1-1-3	1.499	0.516	0.185	0.117	0.050	0.045
BL1-1-4	1.474	0.471	0.110	0.156	0.039	0.027
BL1-1-5	1.542	0.464	0.069	0.129	0.039	0.023
BL1-1-6	1.525	0.457	0.046	0.136	0.034	0.022
BL2-1-2	2.028	0.691	1.986	0.209	1.647	0.937
BL2-1-3	2.241	0.682	1.895	0.201	1.086	0.763
BL2-1-4	2.153	0.644	1.244	0.189	0.561	0.432
BL2-1-5	1.692	0.624	0.973	0.176	0.233	0.225
BL2-1-6	1.469	0.579	0.709	0.163	0.080	0.085

Table 4.13.: Results of *Speed* on *Bikes* and *Trees* sequence. (unit: ms)

Average Distance The following table 4.14 summarizes the results of *average Distance* . In the *Bikes* sequence, BRIEF and ORB are apparently less accuracy than the other descriptors. In the *Trees* sequence there is no significant difference.

avg. Distance	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
BL1-1-2	0.526	0.670	1.119	0.910	0.674	0.703
BL1-1-3	0.629	0.778	1.353	1.179	1.016	1.074
BL1-1-4	0.797	1.004	1.674	1.595	nan	1.310
BL1-1-5	0.973	1.150	1.707	1.718	nan	1.874
BL1-1-6	1.074	1.283	1.670	1.291	nan	nan
BL2-1-2	1.094	1.092	1.402	0.958	0.985	1.225
BL2-1-3	0.928	1.067	1.307	0.909	1.074	1.209
BL2-1-4	1.611	1.343	1.426	1.204	1.291	1.309
BL2-1-5	1.301	1.648	1.660	nan	1.232	1.711
BL2-1-6	nan	nan	1.670	nan	nan	1.738

Table 4.14.: Results of *average Distance* on *Bikes* and *Trees* sequence.

Summary Figure 4.13 shows the visualized comparison results. Under the blur effect, unstructured images lead to a huge number of detected feature points. BRIEF obtains the highest value on Repeatability, Recall and Efficiency measures. With regards to the time, ORB demonstrates the speed benefits, because it limits the maximal number of detected features, while the other descriptors consume huge time to match a large number of processing feature points.

4.5.3. Rotation + Zoom

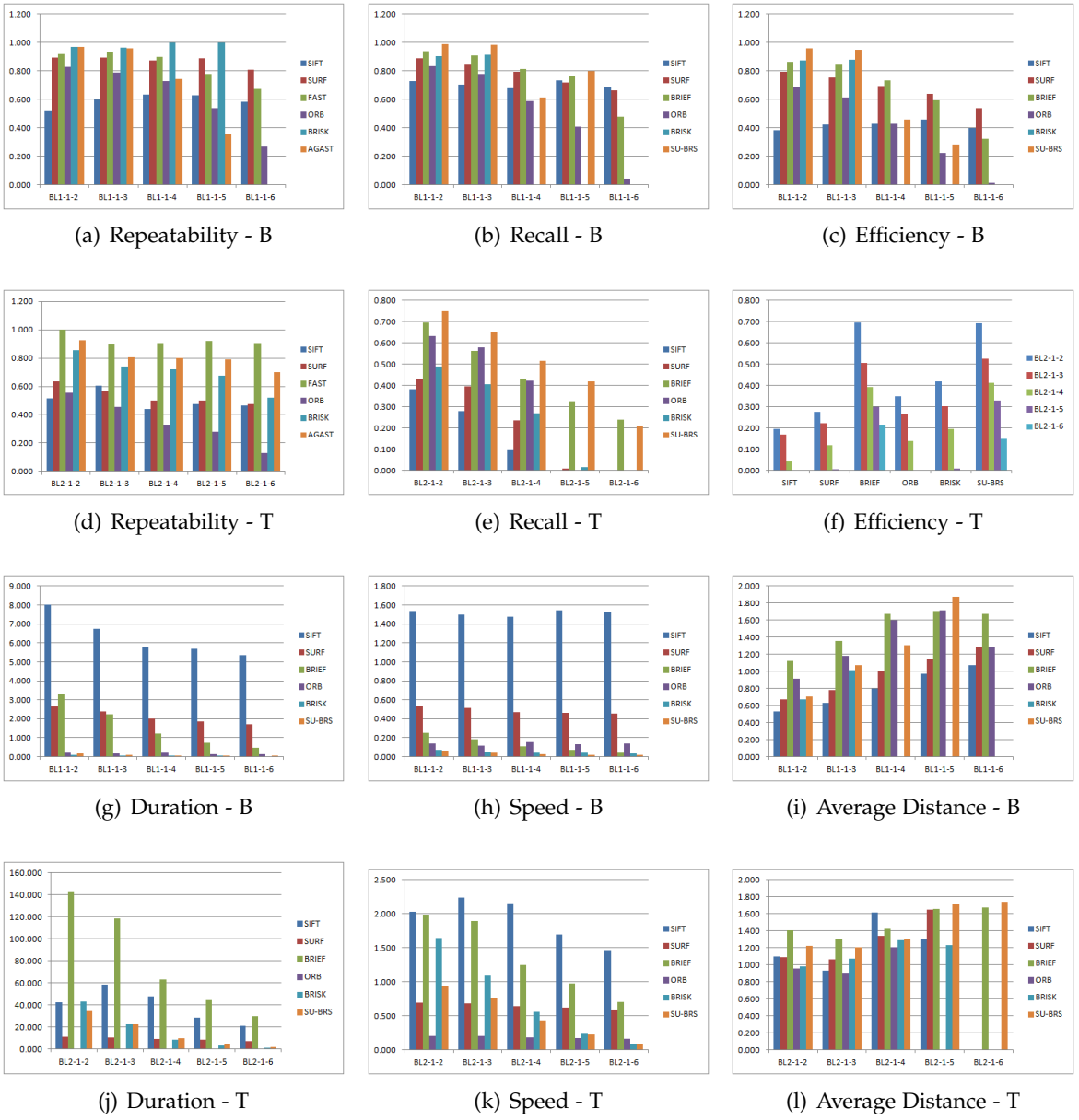
This experiment is designed to evaluate the performance of descriptors on rotated and zoomed images. As introduced in chapter 3, some descriptors are actually not rotation or scale invariant. Figures 4.8 shows the test image sequences. Both sequences involve substantial rotation and scale changes. Comparing the two sequences, there are more features on the *Boat* images than on the *Bark* images.

Keypoints The following table 4.1 summarizes the number of detected feature keypoints in the query image, the number of correspondences and correct matches found after RANSAC. In the *Bark* sequence, only SIFT, SURF and ORB detect sufficient correct features on the first image pair, all descriptors show weak ability to find feature matches on these rotated and zoomed images. Result on the *Bark* images is relatively better. SURF and BRISK perform outstanding, even on the fourth image pair they match features successfully. BRIEF and SU-BRISK are neither rotation invariant nor scale invariant, their poor performance are expected.

Repeatability The following table 4.16 summarizes the value of *Repeatability*. In the *Bark* sequence, all descriptor perform similarly. None of them has a value over 0.4 . In the *Boat* sequence, SIFT and SURF have relatively lower Repeatability value compared to the other

Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
RZ1-1-2	3520	1268	752	2137	849	477	8554	3403	0
RZ1-1-3	4465	463	0	2129	282	0	11354	1915	0
RZ1-1-4	4518	407	0	2826	192	0	14446	1895	0
RZ1-1-5	4055	383	24	2606	136	0	13946	1321	0
RZ1-1-6	4158	88	0	2721	60	1	15972	809	0
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
RZ1-1-2	702	157	44	100	16	0	504	108	0
RZ1-1-3	702	72	0	247	18	0	989	78	0
RZ1-1-4	702	60	0	907	36	0	2332	163	0
RZ1-1-5	702	43	0	916	23	0	2249	111	0
RZ1-1-6	702	17	0	1335	11	0	3159	49	0
Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
RZ2-1-2	7293	3656	2236	5738	3109	1755	20979	14309	2762
RZ2-1-3	5517	2422	1579	3989	1751	650	17849	9946	0
RZ2-1-4	4575	1321	333	2957	938	367	14161	5220	0
RZ2-1-5	4655	863	0	2657	554	133	11910	3017	0
RZ2-1-6	3602	431	0	2215	240	0	14411	2380	0
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
RZ2-1-2	702	527	415	5197	3485	1995	5233	3482	2410
RZ2-1-3	702	446	273	4150	2540	1303	4205	2557	0
RZ2-1-4	702	238	1	2454	1223	388	2468	1249	0
RZ2-1-5	702	135	0	1813	651	36	1833	668	0
RZ2-1-6	702	57	0	2118	434	0	2115	440	0

Table 4.15.: Results of detected keypoints on *Bark* and *Boat* sequence. (N_{qry}: number of detected query Keypoints, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC)



(a) Repeatability - B (b) Recall - B (c) Efficiency - B

(d) Repeatability - T (e) Recall - T (f) Efficiency - T

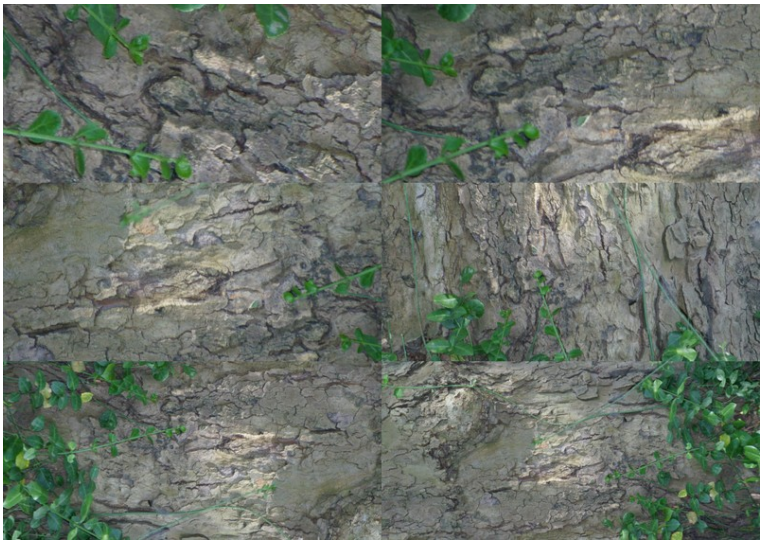
(g) Duration - B (h) Speed - B (i) Average Distance - B

(j) Duration - T (k) Speed - T (l) Average Distance - T

Figure 4.7.: Comparison results on *Bikes* (B) and *Trees* (T) sequence.

descriptors. Comparing the two sequences, all descriptors perform better on the *Boat* images than on the *Bark* images.

Recall The following table 4.17 summarizes the value of *Recall*. The low Recall values of BRIEF and SU-BRISK are expected. In the *Boat* sequence, SURF and BRISK perform best.



(a) *Bark* sequence.

(b) *Boat* sequence.

Figure 4.8.: Test image sequences for Rotation + Zoom - *Bark* and *Boat* sequence.

Comparing the two sequences, all descriptors perform better on the *Boat* images than on the *Bark* images.

Efficiency The following table 4.18 summarizes the value of *Efficiency*. On most image pairs, no correct match is found at all. According to the definition of Efficiency, in such case, no matter how high the value of repeatability is, the value of Efficiency stays 0. Therefore the result of Efficiency is similar to the result of Recall.

Repeatability	SIFT	SURF	FAST	ORB	BRISK	AGAST
RZ1-1-2	0.360	0.397	0.398	0.224	0.160	0.214
RZ1-1-3	0.111	0.132	0.194	0.103	0.073	0.079
RZ1-1-4	0.098	0.085	0.192	0.085	0.125	0.164
RZ1-1-5	0.094	0.060	0.134	0.061	0.080	0.112
RZ1-1-6	0.021	0.027	0.082	0.024	0.038	0.049
RZ2-1-2	0.501	0.552	0.749	0.751	0.740	0.727
RZ2-1-3	0.439	0.439	0.557	0.635	0.612	0.608
RZ2-1-4	0.289	0.317	0.369	0.339	0.498	0.506
RZ2-1-5	0.185	0.209	0.253	0.192	0.359	0.364
RZ2-1-6	0.120	0.108	0.165	0.081	0.205	0.208

Table 4.16.: Results of *Repeatability* on *Bark* and *Boat* sequence.

Recall	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
RZ1-1-2	0.59	0.562	0	0.280	0	0
RZ1-1-3	0.00	0	0	0	0	0
RZ1-1-4	0.00	0	0	0	0	0
RZ1-1-5	0.06	0	0	0	0	0
RZ1-1-6	0.00	0.017	0	0	0	0
RZ2-1-2	0.612	0.564	0.193	0.787	0.572	0.692
RZ2-1-3	0.652	0.371	0	0.612	0.513	0
RZ2-1-4	0.252	0.391	0	0.004	0.317	0
RZ2-1-5	0	0.240	0	0	0.055	0
RZ2-1-6	0	0	0	0	0	0

Table 4.17.: Results of *Recall* on *Bark* and *Boat* sequence.

Efficiency	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
RZ1-1-2	0.214	0.223	0	0.063	0	0
RZ1-1-3	0	0	0	0	0	0
RZ1-1-4	0	0	0	0	0	0
RZ1-1-5	0.006	0	0	0	0	0
RZ1-1-6	0	0.0004	0	0	0	0
RZ2-1-2	0.307	0.312	0.145	0.591	0.424	0.503
RZ2-1-3	0.286	0.163	0	0.389	0.314	0
RZ2-1-4	0.073	0.124	0	0.001	0.158	0
RZ2-1-5	0	0.050	0	0	0.020	0
RZ2-1-6	0	0	0	0	0	0

Table 4.18.: Results of *Efficiency* on *Bark* and *Boat* sequence.

Duration and Speed The following tables 4.19 and 4.20 summarize the result of *Duration* and *Speed* on the *Bark* and *Boat* sequences. The both measure the time consumption. Similar to the results on the *Light* 4.5.1and *Bikes* 4.5.2images, ORB, BRISK and SU-BRISK finish the matching process in the shortest time and SIFT runs most slowly. Compared to the *Bark* sequence, all descriptors require more time on the *Boat* images, but the time incremental of BRIEF and ORB is much smaller than other descriptors.

Duration	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
RZ1-1-2	11.12	1.81	9.48	0.13	0.03	0.08
RZ1-1-3	11.84	1.83	12.89	0.14	0.04	0.13
RZ1-1-4	12.22	2.07	16.24	0.16	0.09	0.28
RZ1-1-5	11.61	1.99	15.49	0.14	0.09	0.27
RZ1-1-6	11.49	2.02	17.72	0.14	0.13	0.43
RZ2-1-2	25.38	6.18	10.48	0.19	1.18	1.08
RZ2-1-3	20.71	4.75	8.92	0.19	0.98	0.88
RZ2-1-4	18.67	4.05	7.17	0.18	0.67	0.58
RZ2-1-5	18.81	3.85	6.12	0.17	0.52	0.46
RZ2-1-6	16.87	3.42	7.37	0.17	0.59	0.50

Table 4.19.: Results of *Duration* on *Bark* and *Boat* sequence.

Speed	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
RZ1-1-2	1.448	0.412	0.514	0.095	0.077	0.051
RZ1-1-3	1.372	0.418	0.607	0.098	0.079	0.065
RZ1-1-4	1.408	0.407	0.668	0.113	0.077	0.084
RZ1-1-5	1.413	0.410	0.650	0.101	0.078	0.083
RZ1-1-6	1.381	0.407	0.685	0.102	0.081	0.104
RZ2-1-2	1.677	0.543	0.262	0.138	0.119	0.108
RZ2-1-3	1.551	0.494	0.242	0.132	0.110	0.098
RZ2-1-4	1.505	0.472	0.216	0.130	0.093	0.080
RZ2-1-5	1.506	0.464	0.197	0.123	0.080	0.069
RZ2-1-6	1.475	0.436	0.220	0.123	0.086	0.073

Table 4.20.: Results of *Speed* on *Bark* and *Boat* sequence.

Average Distance The following table 4.21 summarizes the results of *average Distance*. Although there are only several successful matching result in the table, according to these matches, the result of SIFT features is still more accurate than other descriptors.

Summary Figure 4.9 shows the visualized comparison results. In conclusion, under the rotation and scale changes, image matching process becomes more difficult. The property of

avg. Distance	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
RZ1-1-2	0.794	1.271	nan	1.343	nan	nan
RZ1-1-3	nan	nan	nan	nan	nan	nan
RZ1-1-4	nan	nan	nan	nan	nan	nan
RZ1-1-5	1.487	nan	nan	nan	nan	nan
RZ1-1-6	nan	0.077	nan	nan	nan	nan
RZ2-1-2	0.815	1.499	1.698	1.035	1.133	1.303
RZ2-1-3	0.688	1.712	nan	1.094	1.145	nan
RZ2-1-4	1.792	1.598	nan	0.028	1.632	nan
RZ2-1-5	nan	1.746	nan	nan	1.587	nan
RZ2-1-6	nan	nan	nan	nan	nan	nan

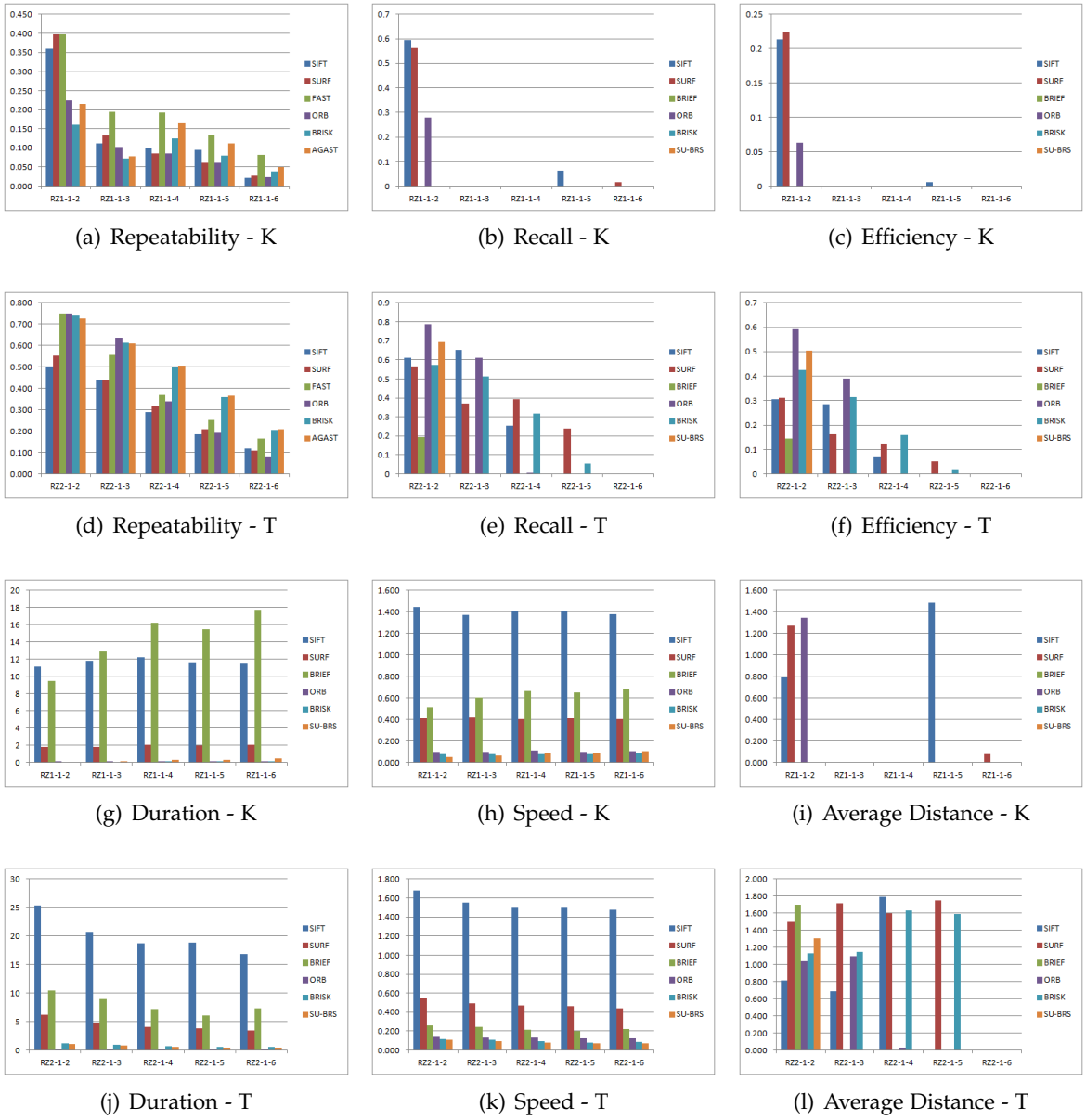
Table 4.21.: Results of *average Distance* on *Bark* and *Boat* sequence.

rotation and scale variant by BRIEF and SU-BRISK descriptors appearances immediately in this case. Among the 4 rotation and scale invariant descriptors, SURF and BRISK perform better than SIFT and ORB in this experiment. With regards to the time consumption, using the binary descriptor ORB and BRISK brings significantly more advantages than vector descriptors SIFT and SURF.

4.5.4. Viewpoint change

In a mobile VO system, camera may often capture images from different viewpoints or camera angles. This section evaluates the performance of the matching process under the condition of viewpoint change. Figures 4.10 shows the two test image sequences. Both sequences involve substantial view angle changes. And since both scenes are plane, the homograhy can still be used to compute the ground truth in this case.

Keypoints The following table 4.1 summarizes the number of detected feature keypoints in the query image, the number of correspondences and correct matches found after RANSAC. On the *Graffiti* images, all descriptors perform poorly on the last three image pairs, only ORB finds 24 correct matches on the image pair 1|4, other descriptors fail to match the features successfully on these images. On the *Wall* images, the number of features is increased significantly compared to the *Graffiti* images. Particularly FAST (detector for BRIEF) detects more than 25000 feature points on the first five images, and BRIEF finds more than 20000 correct matches on the first image pair. Even on the last image pair, which is much more difficult to match because of the substantial view angle change, more than 200 features are matched successfully by BRIEF, which is sufficient for the motion estimation. With the exception of ORB, all descriptors extract much more feature points on the *Wall* images than on the *Graffiti* images, and meanwhile more correct matches are found. The reason may be the similarity of the features. Unlike distinct and unique pattern in the *Graffiti* images, the texture of the *Wall* images is less structured and more repetitive, many features have similar



(a) Repeatability - K (b) Recall - K (c) Efficiency - K

(d) Repeatability - T (e) Recall - T (f) Efficiency - T

(g) Duration - K (h) Speed - K (i) Average Distance - K

(j) Duration - T (k) Speed - T (l) Average Distance - T

Figure 4.9.: Comparison results on *Bark* (K) and *Boat* (T) sequence.

Harris corner measure, after sorting only 702 keypoints remain. Some good features, which may be more suitable for the matching process, are filtered out.

Repeatability The following table 4.23 summarizes the value of *Repeatability* . There is no obvious difference at Repeatability metric between seven feature detectors. Comparing the

Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
VP1-1-2	3172	1197	925	3536	1512	963	6801	3493	147
VP1-1-3	3488	941	531	3831	979	303	7204	3036	0
VP1-1-4	3616	697	0	3498	619	0	8818	2831	0
VP1-1-5	3825	379	0	4163	343	0	9039	1806	0
VP1-1-6	4433	279	0	3610	250	0	12301	1688	0
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
VP1-1-2	702	502	374	1261	638	492	1685	858	392
VP1-1-3	702	369	155	1415	581	95	1976	789	32
VP1-1-4	702	306	24	1405	413	0	2065	606	0
VP1-1-5	702	133	0	1498	252	0	2128	381	0
VP1-1-6	702	99	0	1811	181	0	2921	295	0
Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
VP2-1-2	9690	5447	4225	5956	4335	3063	27704	24774	21407
VP2-1-3	9619	4947	3591	5754	3806	2379	26384	22776	18454
VP2-1-4	9255	3628	2152	5816	2664	1115	27751	19573	9588
VP2-1-5	9687	2980	479	5709	1858	13	27321	17383	3783
VP2-1-6	9601	2035	0	5393	1119	1	27181	13771	214
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
VP2-1-2	702	463	311	1753	1400	933	1749	1379	1329
VP2-1-3	702	435	263	1627	1231	695	1634	1213	1169
VP2-1-4	702	277	123	1862	975	425	1887	954	808
VP2-1-5	702	164	1	1900	735	53	1920	726	436
VP2-1-6	702	82	0	2121	472	0	2130	469	7

Table 4.22.: Results of detected keypoints on *Graffiti* and *Wall* sequence. (N_{qry}: number of detected query Keypoints, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC)

(a) *Graffiti* sequence.

(b) *Wall* sequence.

Figure 4.10.: Test image sequences for viewpoint change - *Graffiti* and *Wall* sequence.

two sequences, Repeatability value for each detector on the *Wall* images is better than this on the *Graffiti* images with the only exception of ORB, which performs slightly worse on the *Wall* images than on the *Graffiti* images.

Recall The following table 4.24 summarizes the value of *Recall* . On the *Graffiti* images, ORB obtains the highest Recall value on the first three image pairs. Almost all Recall values on the last two image pairs are 0, because no correct matches are found on these images. On

Repeatability	SIFT	SURF	FAST	ORB	BRISK	AGAST
VP1-1-2	0.435	0.472	0.575	0.715	0.690	0.685
VP1-1-3	0.342	0.306	0.500	0.526	0.628	0.630
VP1-1-4	0.254	0.193	0.466	0.436	0.446	0.484
VP1-1-5	0.138	0.107	0.297	0.189	0.272	0.304
VP1-1-6	0.101	0.078	0.278	0.141	0.196	0.235
VP2-1-2	0.651	0.728	0.894	0.660	0.799	0.788
VP2-1-3	0.591	0.661	0.863	0.620	0.757	0.742
VP2-1-4	0.433	0.458	0.705	0.395	0.524	0.506
VP2-1-5	0.356	0.325	0.636	0.234	0.387	0.378
VP2-1-6	0.243	0.207	0.507	0.117	0.223	0.220

Table 4.23.: Results of repeatability on the *Graffiti* and *Wall* sequences.

the *Wall* sequence, all descriptors obtain a high Recall value on the first three image pairs, BRIEF and SU-BRISK descriptor perform best, even on the last image pair, they have a Recall value over 0.015. Comparing the two sequences, Recall values on the *Wall* images are better than this on the *Graffiti* images.

Recall	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
VP1-1-2	0.773	0.637	0.042	0.745	0.771	0.457
VP1-1-3	0.564	0.310	0	0.420	0.164	0.041
VP1-1-4	0	0	0	0.078	0	0
VP1-1-5	0	0	0	0	0	0
VP1-1-6	0	0	0	0	0	0
VP2-1-2	0.776	0.707	0.864	0.672	0.666	0.964
VP2-1-3	0.726	0.625	0.810	0.605	0.565	0.964
VP2-1-4	0.593	0.419	0.490	0.444	0.436	0.847
VP2-1-5	0.161	0.007	0.218	0.006	0.072	0.601
VP2-1-6	0	0.001	0.016	0	0	0.015

Table 4.24.: Results of *Recall* on *Graffiti* and *Wall* sequence.

Efficiency The following table 4.25 summarizes the value of *Efficiency* . This result is similar to the Recall metric, BRIEF and SU-BRISK have the best performance.

Duration and Speed The following tables 4.26 and 4.27 summarize the result of time measure - *Duration* and *Speed* . The time advantage of using binary description is already discussed in section 4.5.3.

Efficiency	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
VP1-1-2	0.336	0.301	0.024	0.533	0.532	0.313
VP1-1-3	0.193	0.095	0	0.221	0.103	0.026
VP1-1-4	0	0	0	0.034	0	0
VP1-1-5	0	0	0	0	0	0.000
VP1-1-6	0	0	0	0	0	0
VP2-1-2	0.505	0.514	0.773	0.443	0.532	0.760
VP2-1-3	0.429	0.413	0.699	0.375	0.427	0.715
VP2-1-4	0.257	0.192	0.346	0.175	0.228	0.428
VP2-1-5	0.057	0.002	0.138	0.001	0.028	0.227
VP2-1-6	0	0.0002	0.008	0	0	0.003

Table 4.25.: Results of *Efficiency* on *Graffiti* and *Wall* sequence.

Duration	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
VP1-1-2	8.72	3.85	6.03	0.15	0.42	0.26
VP1-1-3	8.82	4.54	5.76	0.15	0.54	0.31
VP1-1-4	9.13	3.38	7.14	0.16	0.46	0.31
VP1-1-5	9.80	3.75	8.04	0.16	0.61	0.36
VP1-1-6	11.57	3.92	10.09	0.16	0.69	0.77
VP2-1-2	34.07	7.22	26.50	0.24	0.41	0.31
VP2-1-3	34.77	7.00	25.13	0.23	0.37	0.30
VP2-1-4	33.63	7.39	26.35	0.23	0.40	0.33
VP2-1-5	34.22	7.12	26.77	0.23	0.40	0.35
VP2-1-6	33.21	6.52	25.89	0.23	0.45	0.39

Table 4.26.: Results of Duration on *Graffiti* and *Wall* sequence. (unit: s)

Speed	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
VP1-1-2	1.473	0.571	0.468	0.106	0.192	0.088
VP1-1-3	1.415	0.645	0.434	0.109	0.230	0.095
VP1-1-4	1.434	0.504	0.479	0.112	0.199	0.095
VP1-1-5	1.490	0.509	0.532	0.112	0.254	0.108
VP1-1-6	1.611	0.575	0.549	0.111	0.252	0.186
VP2-1-2	1.887	0.551	0.410	0.169	0.083	0.064
VP2-1-3	1.932	0.542	0.397	0.167	0.077	0.062
VP2-1-4	1.908	0.571	0.407	0.164	0.079	0.065
VP2-1-5	1.895	0.554	0.417	0.165	0.079	0.069
VP2-1-6	1.848	0.520	0.404	0.164	0.085	0.073

Table 4.27.: Results of *Speed* on *Graffiti* and *Wall* sequence. (unit: ms)

Average Distance The following table 4.28 summarizes the value of *average Distance* . As in the previous experiments, SIFT has the smallest position error.

avg. Distance	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
VP1-1-2	0.870	1.425	1.784	1.051	0.860	1.107
VP1-1-3	1.076	1.739	nan	1.286	1.636	1.297
VP1-1-4	nan	nan	nan	1.587	nan	nan
VP1-1-5	nan	nan	nan	nan	nan	nan
VP1-1-6	nan	nan	nan	nan	nan	nan
VP2-1-2	0.388	0.760	0.873	0.802	0.618	0.650
VP2-1-3	0.587	1.043	1.016	0.917	0.725	0.794
VP2-1-4	0.775	1.335	1.163	1.016	0.892	0.915
VP2-1-5	1.379	1.411	1.445	0.378	1.635	1.055
VP2-1-6	nan	0.301	1.907	nan	nan	1.246

Table 4.28.: Results of *average Distance* on *Graffiti* and *Wall* sequence.

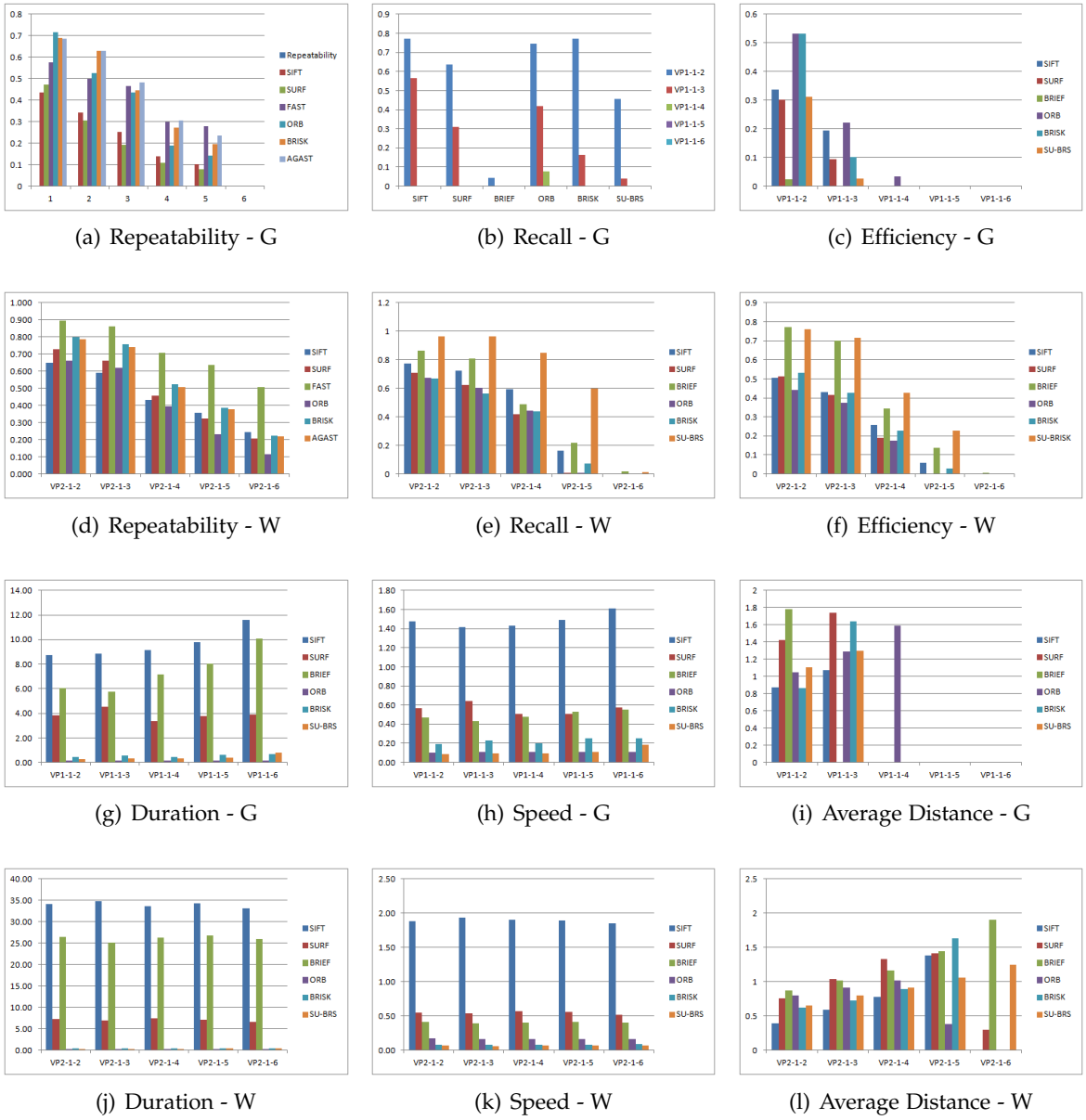
Summary Figure 4.11 shows the visualized comparison results. When the viewpoint or camera angle are substantial changed, it is hard to match features on the image pair. The more feature points that are detected, the better the matching results. On *Graffiti* images, ORB shows the best performance, the combination of FAST and BRIEF can only find few correct matches on the first image pair. But on *Wall* images, BRIEF and SU-BRISK show outstanding performance. Comparing the time consumption among this three descriptors, ORB and SU-BRISK run faster than BRIEF.

4.5.5. JPEG compression

In some case, image compression is required in the post-processing. Although we will in the following assume that VO applications do not involve this step, it is included in the program for comparison purpose. Figures 4.12 shows the test image sequence.

Keypoints The following table 4.1 summarizes the number of detected feature keypoints in the query image, the number of correspondences and correct matches found after RANSAC. Same as the results on previous image sequences, the combination of FAST and BRIEF detects the most number of features, correspondences and correct matches. All descriptors find some correct matches successfully on all six image pairs.

Repeatability The following table 4.30 summarizes the value of *Repeatability* on this image sequence. FAST and ORB detector perform better than other detectors, even on the worst image pair 1|6, the Repeatability values retain above 0.75. SIFT detector performs poorly on this sequence.



(a) Repeatability - G (b) Recall - G (c) Efficiency - G

(d) Repeatability - W (e) Recall - W (f) Efficiency - W

(g) Duration - G (h) Speed - G (i) Average Distance - G

(j) Duration - W (k) Speed - W (l) Average Distance - W

Figure 4.11.: Comparison results on *Graffiti* (G) and *Wall* (W) sequence.

Recall The following table 4.31 summarizes the value of *Recall* on this sequence. With the only exception of SIFT descriptor, all descriptors show outstanding performance on this metric. And among all six descriptors, SU-BRISK descriptor performs best, the Recall value on the first three image pairs exceeds 0.9.

Keyp.	SIFT			SURF			BRIEF		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
CMP-1-2	5732	3132	2461	3898	3579	3130	18757	17313	16193
CMP-1-3	6795	3063	2052	3888	3329	2813	17882	16158	14874
CMP-1-4	6626	2693	1509	3812	2990	2325	14321	12706	11176
CMP-1-5	4539	1873	794	3595	2371	1503	8423	7461	5927
CMP-1-6	2722	1124	145	3197	1728	957	3151	2806	2090
Keyp.	ORB			BRISK			SU-BRS		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
CMP-1-2	702	681	642	3760	3361	3126	6013	5286	5179
CMP-1-3	702	667	603	3640	3103	2778	5921	4919	4757
CMP-1-4	702	662	575	3539	2784	2281	5669	4335	4109
CMP-1-5	702	602	456	3209	2208	1492	4748	3359	3000
CMP-1-6	702	528	355	3143	1823	894	6870	3446	2232

Table 4.29.: Results of detected keypoints on *Jpg* sequence. (N_{qry}: number of detected query Keypoints, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC)

Repeatability	SIFT	SURF	FAST	ORB	BRISK	AGAST
CMP-1-2	0.694	0.918	0.923	0.970	0.894	0.879
CMP-1-3	0.679	0.856	0.904	0.950	0.852	0.831
CMP-1-4	0.597	0.784	0.887	0.943	0.787	0.765
CMP-1-5	0.415	0.660	0.886	0.858	0.688	0.707
CMP-1-6	0.413	0.541	0.891	0.752	0.580	0.541

Table 4.30.: Results of *Repeatability* on *Jpg* sequence.

Recall	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
CMP-1-2	0.786	0.875	0.935	0.943	0.930	0.980
CMP-1-3	0.670	0.845	0.921	0.904	0.895	0.967
CMP-1-4	0.560	0.778	0.880	0.869	0.819	0.948
CMP-1-5	0.424	0.634	0.794	0.757	0.676	0.893
CMP-1-6	0.129	0.554	0.745	0.672	0.490	0.648

Table 4.31.: Results of *Recall* on *Jpg* sequence.

Figure 4.12.: Test image sequences for JPEG compression - *Jpg* sequence.

Efficiency The following table 4.32 summarizes the value of *Efficiency* on this sequence. Combining the both Repeatability and Recall metrics, SIFT shows the worst performance, its result is much lower than any other descriptors, on the last image pair the value is only 0.053. ORB and BRIEF perform best among all descriptors on this quality measure.

Efficiency	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
CMP-1-2	0.545	0.803	0.863	0.915	0.831	0.861
CMP-1-3	0.455	0.724	0.832	0.859	0.763	0.803
CMP-1-4	0.334	0.610	0.780	0.819	0.645	0.725
CMP-1-5	0.176	0.418	0.704	0.650	0.465	0.632
CMP-1-6	0.053	0.299	0.663	0.506	0.284	0.351

Table 4.32.: Results of *Efficiency* on *Jpg* sequence.

Duration The following table 4.33 summarizes the results of *Duration* on this image sequence. FAST (detector used for BRIEF) detects a lot more features than any other detectors. On the first image pair, more than 17000 feature points are found by SIFT, the processing time of 40s for BRIEF features is far too long. Comparing the result of BRIEF on the *Boat* images in section 4.5.3 (Keypoints) and section 4.5.3 (Duration), BRIEF finished the matching process for more than 20000 features in 10.48 seconds on the *Boat* image. Among all six descriptors, ORB requires the shortest time, only 0.18 second for each image pair.

Duration	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
CMP-1-2	17.71	3.87	40.16	0.18	4.84	4.46
CMP-1-3	17.37	4.31	38.33	0.18	4.49	3.93
CMP-1-4	16.12	3.78	32.18	0.18	4.47	3.82
CMP-1-5	13.33	3.73	24.05	0.18	4.07	3.20
CMP-1-6	10.68	3.40	7.32	0.17	3.46	4.64

Table 4.33.: Results of *Duration* on *Jpg sequence.*

Speed The following table 4.34 summarizes the value of *Speed* on this sequence. SIFT requires around 1.5 second to process one single feature, much more than any other approaches. ORB runs fastest, requires only 0.12 second for each feature.

Speed	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
CMP-1-2	1.728	0.495	1.066	0.131	0.609	0.360
CMP-1-3	1.536	0.552	1.042	0.128	0.574	0.320
CMP-1-4	1.447	0.488	0.968	0.126	0.579	0.318
CMP-1-5	1.473	0.496	0.880	0.125	0.550	0.288
CMP-1-6	1.475	0.477	0.332	0.123	0.472	0.351

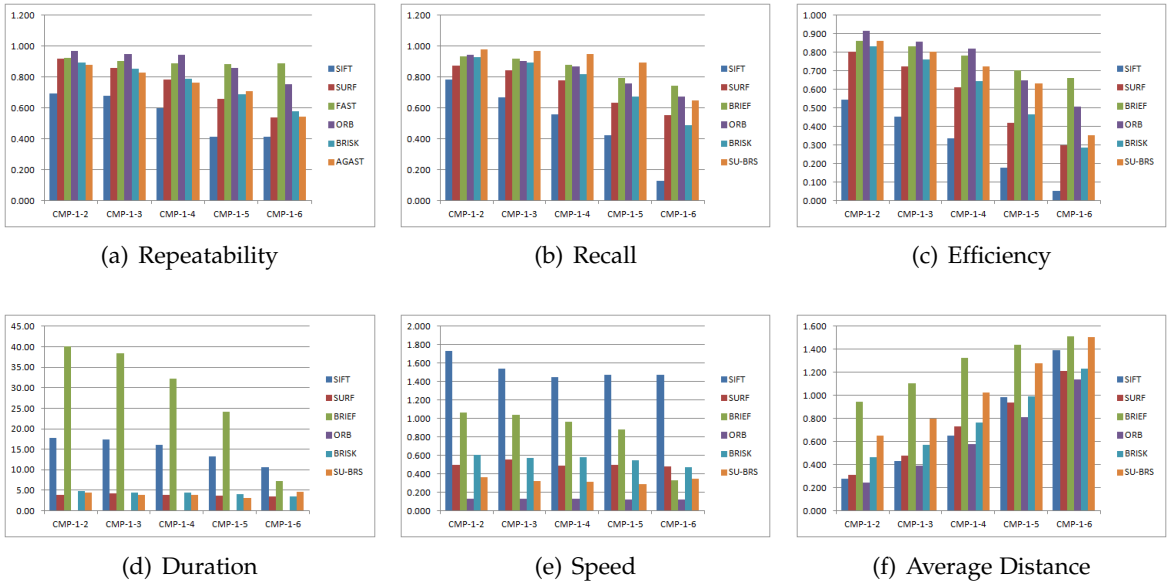
Table 4.34.: Results of *Speed* on *Jpg* sequence.

Average Distance The following table 4.35 summarizes the result of *average Distance* on this sequence. The matching result of ORB is most accurate, even slightly better than SIFT, which has the lowest average Distance on all previous image sequences. The position error of BRIEF is significantly bigger than other descriptors.

avg. Distance	SIFT	SURF	BRIEF	ORB	BRISK	SU-BRS
CMP-1-2	0.280	0.314	0.945	0.245	0.462	0.649
CMP-1-3	0.433	0.476	1.103	0.390	0.574	0.800
CMP-1-4	0.650	0.729	1.324	0.575	0.765	1.027
CMP-1-5	0.983	0.940	1.437	0.814	0.988	1.278
CMP-1-6	1.395	1.215	1.510	1.137	1.229	1.508

Table 4.35.: Results of *average Distance* on *Jpg* sequence.

Summary Figure 4.9 shows the visualized comparison results on this sequence. All descriptors perform well on this sequence. ORB shows the best performance on almost all metrics, not only the quality measures but also on time consumption and error evaluation metrics. BRIEF performs notably worse than any other approaches on this sequence.



(a) Repeatability (b) Recall (c) Efficiency

(d) Duration (e) Speed (f) Average Distance

Figure 4.13.: Comparison results on *Jpg* sequence.

4.6. Conclusion

Under different conditions the relative performance of the descriptors is quite different.

Illumination Change BRIEF obtains the highest value on quality measure, but because of a large amount of detected keypoints, loses the superiority on time consumption. Since there is no rotation and scale change in the data set, the rotation and scale invariant ORB and BRISK are at a disadvantage, and are expected to perform worse than BRIEF and SU-BRISK. The time advantage of using binary description comparing to vector description is demonstrated vividly on this image sequence.

Blur BRIEF obtains the highest value on Repeatability, Recall and Efficiency metrics. ORB demonstrates the speed benefits through limiting the maximal number of detected features, while the other descriptors consume huge time to match a large number of processing feature points.

Rotation and Zoom Under rotation and scale change, the image matching process becomes more difficult. BRIEF and SU-BRISK descriptors show their nature of rotation and scale variant immediately. Among the 4 rotation and scale invariant descriptors, SURF and BRISK perform better than SIFT and ORB. With regards to the time consumption, using a binary

descriptor such as ORB and BRISK provides significant advantage over the vector descriptors SIFT and SURF.

Viewpoint Change When the viewpoint or camera angle is substantially changed, it is hard to match features between a pair of images. ORB, BRIEF and SU-BRISK show the outstanding performance in this case. Comparing the time consumption among these three descriptors, ORB and SU-BRISK run faster than BRIEF.

JPEG Compression ORB shows the best performance when the test images are compressed. In contras, BRIEF does not show the outstanding performance like in the previous experiments of illumination, blur and viewpoint changes, it performs obviously worse than any other approaches in this case.

Descriptor suggestion for VO-Application Considering all conclusions obtained in this chapter, ORB and BRISK are the best choice for a mobile VO-System. Disregarding JPEG compression, both perform well under all aforementioned conditions of change. If it is expected in advance that the image sequence has no rotation and scale deformation at all, ORB and SU-BRISK are suggested. In the case of the fixed camera position, BRIEF and SU-BRISK are the first choice.

5. Experimental evaluation

In the last chapter several experiments are finished to compare the performance of feature descriptors under different conditions. Now the focus is drifted to improvement, trying to enhance the performance of image matching process. Three experiments are designed for this purpose: using lens with different focal length and saving image with different bit depth, adjusting the parameters and applying cross check filter. After that, another experiment is executed to analyze the time consumption of each step in image matching process.

5.1. Varying focal length and bit depth

This experiment is trying to improve the image quality by using lens with varying focal length and saving in varying bit depth before image processing. In this experiment, only 4 descriptors are tested, they are SIFT, SURF, BRIEF and ORB. All test images used are captured local in a office. There are three image sequences:

1. *4-2mm* : camera uses lens with 4.2 mm focal length and images are saved with 8 bit depth.
2. *4-2mm-12bit* sequence : camera uses lens with 4.2 mm focal length and images are saved with 12 bit depth.
3. *6mm* sequence : camera uses lens with 6 mm focal length and images are saved with 8 bit depth.

Figure 5.1 shows the first and third test image sequence. Same as the dataset used in last chapter, each sequence contains 6 images. Because of the environment of this experiment, it is hard to capture the images with gradually reducing illumination. The lighting condition of the first three images are better than the last three one. The first image is used as the reference image.

Keypoints The following table 5.1 summarizes the number of detected feature keypoints in query image, correspondences and correct matches found after RANSAC on three test sequences. Because of the poor illumination condition on the last three image pairs in each sequence, the number of detect features are obviously reduced compare to the first two image pairs. Comparing the *4-2mm* and *4-2-mm-12-bit* sequence, after increasing the bit depth, the number of detected and matches features by SIFT, SURF and BRIEF is also increased, but this number of ORB stays almost the same. Comparing the *4-2mm* and *6mm* sequence, the

(a) 4.2mm

(b) 6mm

Figure 5.1.: Test images of *4-2mm* and *6mm* sequence.

result of SIFT, SURF and BRIEF does not change a lot. At ORB descriptor, with increasing of focal length, the number of detected and matches features are significantly increased on the first two image pairs but decreased on the last three image pairs.

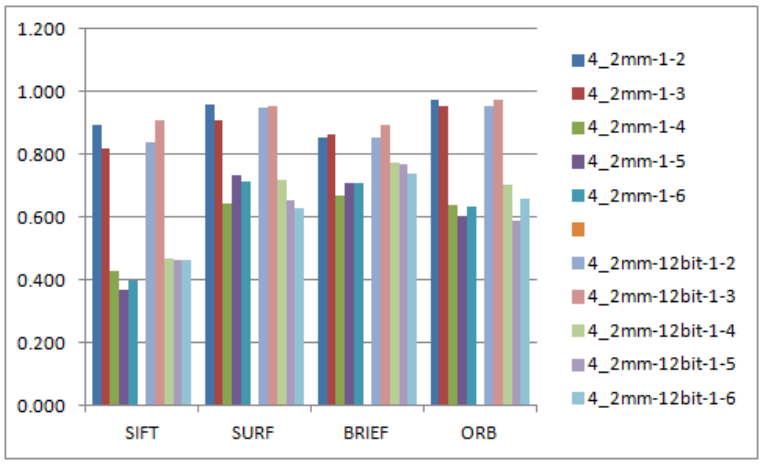
Repeatability The following table 5.2 summarizes the value of *Repeatability* in this experiment and the figure 5.2 visualizes this result. Because of the poor illumination condition in last three image pairs in each sequence, the detectors shows worse performance on last three image pairs than the first two image pairs. Comparing the *4-2mm* and *4-2-mm-12-bit* sequence, changing of bit depth has almost no influence on the Repeatability value. Comparing the the *4-2mm* and *6mm* sequence, with increasing of focal length, the Repeatability value is obviously increased on the the last three image pairs with poor illumination.

Keypoints	SIFT			SURF			BRIEF			ORB		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
4-2mm-1-2	173	155	149	281	269	266	440	374	365	289	282	267
4-2mm-1-3	169	138	130	260	236	222	410	354	334	274	262	236
4-2mm-1-4	14	6	5	31	20	16	57	38	25	39	25	14
4-2mm-1-5	19	7	4	30	22	18	55	39	27	38	23	13
4-2mm-1-6	15	6	5	28	20	16	55	39	25	38	24	15
4-2mm-12bit-1-2	165	138	125	264	250	234	418	356	334	277	264	237
4-2mm-12bit-1-3	162	147	138	282	269	257	423	377	365	299	291	280
4-2mm-12bit-1-4	15	7	5	25	18	13	53	41	30	34	24	13
4-2mm-12bit-1-5	13	6	4	29	19	15	52	40	28	39	23	7
4-2mm-12bit-1-6	13	6	3	27	17	14	57	42	34	38	25	16
6mm-1-2	187	161	154	295	271	248	485	402	347	360	339	284
6mm-1-3	244	221	214	365	341	327	512	441	424	415	405	374
6mm-1-4	16	12	12	28	24	23	54	44	38	16	14	12
6mm-1-5	17	13	13	28	24	21	52	43	37	19	16	12
6mm-1-6	19	12	12	28	24	22	51	40	34	18	16	13

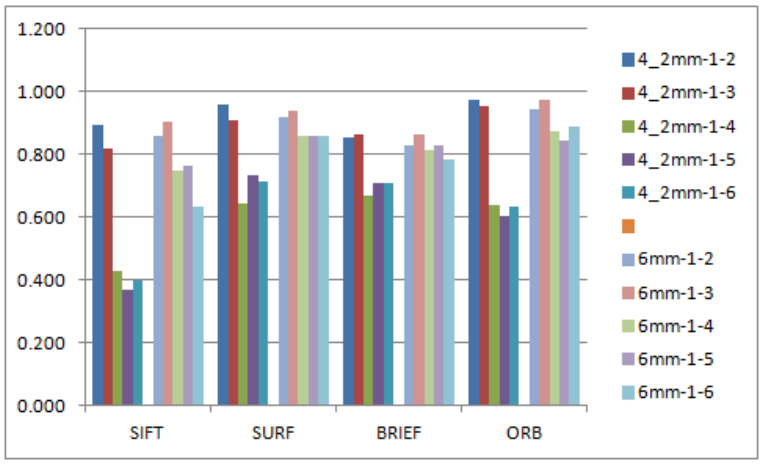
Table 5.1.: Results of detected keypoints using varying focal length and bit depth. (N_{qry}: number of detected query Keypoints, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC)

Repeatability	SIFT	SURF	BRIEF	ORB
4-2mm-1-2	0.896	0.957	0.856	0.976
4-2mm-1-3	0.817	0.908	0.863	0.956
4-2mm-1-4	0.429	0.645	0.667	0.641
4-2mm-1-5	0.368	0.733	0.709	0.605
4-2mm-1-6	0.400	0.714	0.709	0.632
4-2mm-12bit-1-2	0.836	0.947	0.852	0.953
4-2mm-12bit-1-3	0.907	0.954	0.891	0.973
4-2mm-12bit-1-4	0.467	0.720	0.774	0.706
4-2mm-12bit-1-5	0.462	0.655	0.769	0.590
4-2mm-12bit-1-6	0.462	0.630	0.737	0.658
6mm-1-2	0.861	0.919	0.829	0.942
6mm-1-3	0.906	0.939	0.861	0.976
6mm-1-4	0.750	0.857	0.815	0.875
6mm-1-5	0.765	0.857	0.827	0.842
6mm-1-6	0.632	0.857	0.784	0.889

Table 5.2.: Results of *Repeatability* using varying focal length and bit depth



(a) 8 bit vs 12 bit



(b) 4.2 mm vs 6 mm

Figure 5.2.: Comparison results of *Repeatability* using varying focal length and bit depth.

Recall The following table 5.3 summarizes the value of *Recall* in this experiment and Figure 5.3 visualizes this result. In each sequence, the Recall value on the first two image pairs is higher then this on the last three image pairs. The only exception is SIFT on *6mm* sequence, where SIFT finds all correspondences successfully on the last three image pairs. Comparing the *4-2mm* and *4-2-mm-12-bit* sequence, there is no big difference on Recall value. Comparing the the *4-2mm* and *6mm* sequence, with increasing of focal length, the Recall value is obviously increased on the last three image pairs with poor illumination. But there is no significantly change on the first two image pairs.

Duration The following table 5.4 summarizes the result of *Duration* in this experiment and the figure 5.4 visualizes this result. Comparing the *4-2mm* and *4-2-mm-12-bit* sequence, there

Recall	SIFT	SURF	BRIEF	ORB
4-2mm-1-2	0.961	0.989	0.976	0.947
4-2mm-1-3	0.942	0.941	0.944	0.901
4-2mm-1-4	0.833	0.800	0.658	0.560
4-2mm-1-5	0.571	0.818	0.692	0.565
4-2mm-1-6	0.833	0.800	0.641	0.625
4-2mm-12bit-1-2	0.906	0.936	0.938	0.898
4-2mm-12bit-1-3	0.939	0.955	0.968	0.962
4-2mm-12bit-1-4	0.714	0.722	0.732	0.542
4-2mm-12bit-1-5	0.667	0.789	0.700	0.304
4-2mm-12bit-1-6	0.500	0.824	0.810	0.640
6mm-1-2	0.957	0.915	0.863	0.838
6mm-1-3	0.968	0.959	0.961	0.923
6mm-1-4	1	0.958	0.864	0.857
6mm-1-5	1	0.875	0.860	0.750
6mm-1-6	1	0.917	0.850	0.813

Table 5.3.: Results of *Recall* using varying focal length and bit depth

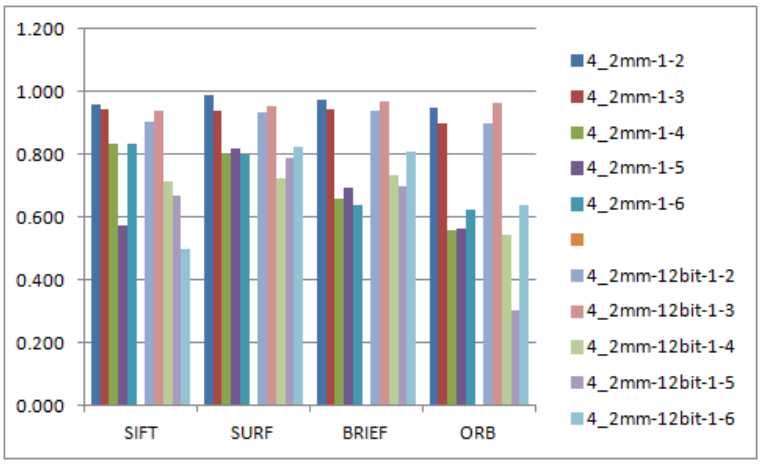
is no significant difference. Comparing the the *4-2mm* and *6mm* sequence, with increasing of focal length, all descriptor require more time to finish the image matching process on the the last three image pairs.

Conclusion Saving the image in more bit depth (from 8 bit to 12 bit), the number of detected feature is reduced, but it does not help to improve the performance for the whole image matching process. Using lens with larger focal length (from 4.2mm to 6 mm), the performance under strong illumination has no change, the performance under weak illumination is significantly enhanced.

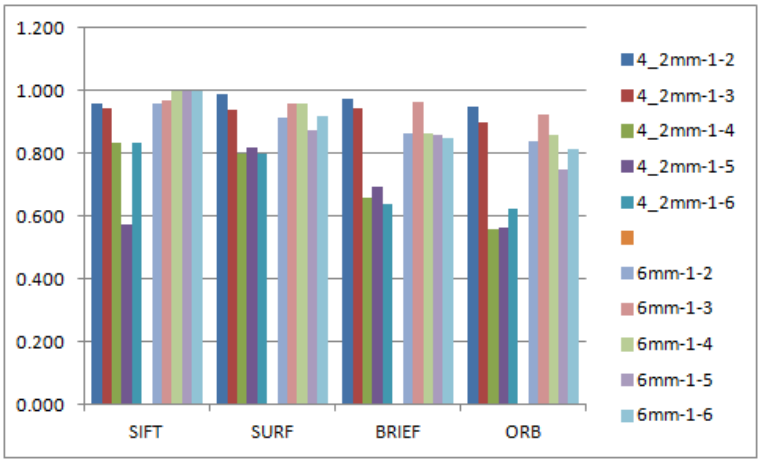
5.2. Varying T in ORB

ORB has showed a outstanding performance in the last chapter, particularly when rotation, scale and viewpoint change is involved. This experiment is trying to improve the ORB matching process by adjusting the parameter T (the threshold used in FAST detection stage, see section 3.1.5). The image sequence used in this experiment is *4-2mm* sequence, which comes from the previous focal length and bit depth experiment. The default value of T is 20, the test interval of T is from 5 to 40, and takes one experimental point every 5 value.

Keypoints The following table 5.5 summarizes the number of detected feature keypoints in query image, correspondences and correct matches found after RANSAC with different



(a) 8 bit vs 12 bit



(b) 4.2 mm vs 6 mm

Figure 5.3.: Visualized *Recall* results using varying focal length and bit depth

value of T. Note that, With increasing T, the number of detected ORB features is reduced. On the image pair 1 | 2, when t=5 and t=10 the number of detected feature point on query image is 702, which is the keypoints limitation of ORB. It means that there could be more than 702 features on the image, after sorting based on Harris corner measure only 702 features are retained. On the last three image pairs, when T > 30, almost no correct matches are found.

Repeatability The following table 5.6 summarizes the value of *Repeatability* in this experiment and the figure 5.5 visualizes this result. The Repeatability value change on the first two image pairs is less evident than this on the last three image pairs. On the last three image pairs, the general tend of Repeatability is to increase until t = 10, then decrease until t = 20,

Duration	SIFT	SURF	BRIEF	ORB
4-2mm-1-2	1.710	0.544	0.046	0.078
4-2mm-1-3	1.694	0.544	0.047	0.062
4-2mm-1-4	1.570	0.513	0.016	0.062
4-2mm-1-5	1.616	0.497	0.015	0.047
4-2mm-1-6	1.647	0.482	0.015	0.063
4-2mm-12bit-1-2	1.741	0.560	0.046	0.062
4-2mm-12bit-1-3	1.741	0.544	0.047	0.078
4-2mm-12bit-1-4	1.570	0.498	0.016	0.062
4-2mm-12bit-1-5	1.601	0.482	0.031	0.062
4-2mm-12bit-1-6	1.601	0.482	0.031	0.062
6mm-1-2	1.835	0.577	0.063	0.093
6mm-1-3	1.881	0.639	0.062	0.094
6mm-1-4	1.725	0.577	0.031	0.062
6mm-1-5	1.819	0.608	0.032	0.078
6mm-1-6	1.808	0.577	0.031	0.063

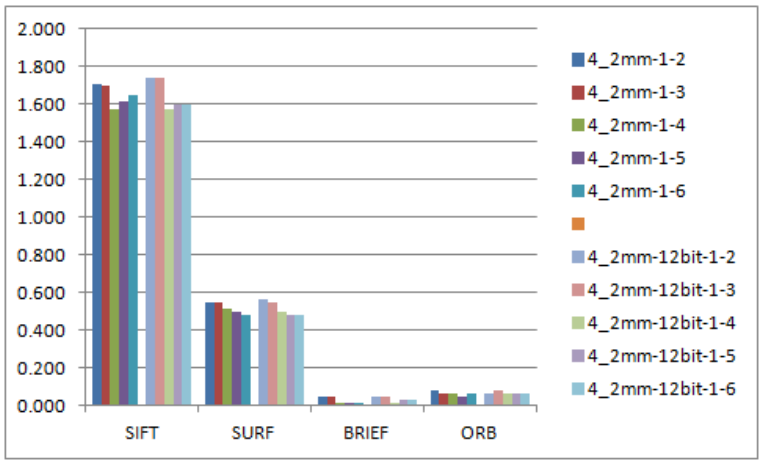
Table 5.4.: Results of *Duration* using varying focal length and bit depth

	IDA-1-2			IDA-1-3			IDA-1-4			IDA-1-5			IDA-1-6		
	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}	N_{qry}	N_{csp}	N_{crt}
t=5	702	681	634	702	667	592	400	233	110	414	235	109	396	232	114
t=10	702	677	631	702	663	590	133	105	62	132	100	58	128	100	58
t=15	564	550	514	534	514	452	73	51	29	70	51	34	68	48	30
t=20	289	282	267	274	262	236	39	25	14	38	23	13	38	24	15
t=25	176	173	165	162	158	143	22	18	6	16	14	7	21	18	11
t=30	113	108	102	101	98	89	12	11	1	12	11	1	14	13	4
t=35	57	56	54	46	42	40	8	7	1	8	7	0	8	7	1
t=40	28	27	27	19	19	19	6	6	0	6	6	0	6	6	0

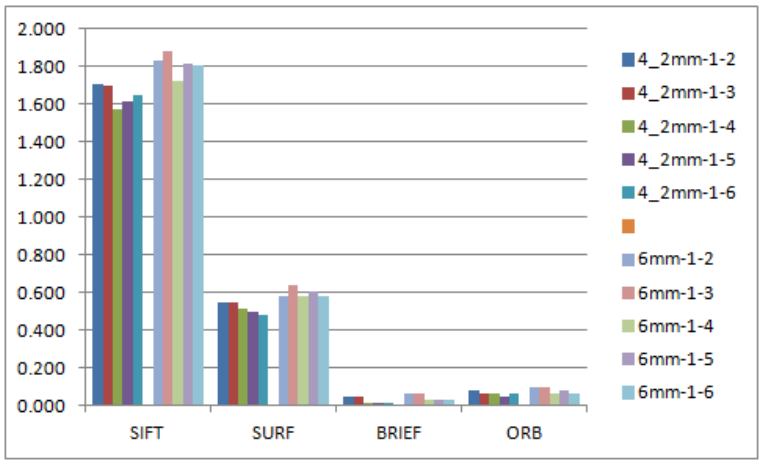
Table 5.5.: Results of detected keypoints with varying T in ORB.

then increase again until the end. The local minimal point appears at t = 20, exactly the default value of T used in ORB.

Recall The following table 5.7 summarizes the value of *Recall* in this experiment and Figure 5.6 visualizes this result. The Recall value changes only slightly on the first two image pairs. On the last three image pairs, with increasing value of T, the general tend of Recall is to increase first at small value of T and than to decrease at large value of T. The inflection point appears between t=15 and t=25. And the maximal value of Recall appears between t=10 and t=20.



(a) 8 bit vs 12 bit



(b) 4.2 mm vs 6 mm

Figure 5.4.: Visualized *Duration* results using varying focal length and bit depth

Duration The following table 5.8 summarizes the result of *Duration* in this experiment and Figure 5.7 visualizes this result. With increasing of T, the processing time is gradually reduced. After t=30, the change becomes slightly.

Conclusion Under the condition of strong illumination, increasing the value of T makes the detected features more cornerness and robustness, therefore the possibility to find the correct matches is increased. Under the condition of inadequate illumination, increasing the value of T causes less points being detected as features, in worst case there are no enough feature for matching process. Actually T=10 and T=15 are more suitable for this *4-2mm* sequence than T=20. But considering all measures, T=20 is a appropriate default value for all kind of images.

Repeatability	IDA-1-2	IDA-1-3	IDA-1-4	IDA-1-5	IDA-1-6
t=5	0.970	0.950	0.583	0.568	0.586
t=10	0.964	0.944	0.789	0.758	0.781
t=15	0.975	0.963	0.699	0.729	0.706
t=20	0.976	0.956	0.641	0.605	0.632
t=25	0.983	0.975	0.818	0.875	0.857
t=30	0.956	0.970	0.917	0.917	0.929
t=35	0.982	0.913	0.875	0.875	0.875
t=40	0.964	1	1	1	1

Table 5.6.: Results of *Repeatability* with varying T in ORB.

Recall	IDA-1-2	IDA-1-3	IDA-1-4	IDA-1-5	IDA-1-6
t=5	0.931	0.888	0.472	0.464	0.491
t=10	0.932	0.890	0.590	0.580	0.580
t=15	0.935	0.879	0.569	0.667	0.625
t=20	0.947	0.901	0.560	0.565	0.625
t=25	0.954	0.905	0.333	0.500	0.611
t=30	0.944	0.908	0.091	0.091	0.308
t=35	0.964	0.952	0.143	0.000	0.143
t=40	1	1	0	0	0

Table 5.7.: Results of *Recall* with varying T in ORB.

Duration	IDA-1-2	IDA-1-3	IDA-1-4	IDA-1-5	IDA-1-6
t=5	0.127	0.134	0.119	0.108	0.110
t=10	0.113	0.125	0.089	0.079	0.093
t=15	0.101	0.097	0.091	0.080	0.079
t=20	0.093	0.098	0.081	0.094	0.095
t=25	0.076	0.090	0.074	0.092	0.086
t=30	0.083	0.089	0.082	0.077	0.078
t=35	0.080	0.088	0.064	0.078	0.079
t=40	0.074	0.085	0.085	0.086	0.083

Table 5.8.: Results of *Duration* with varying T in ORB.

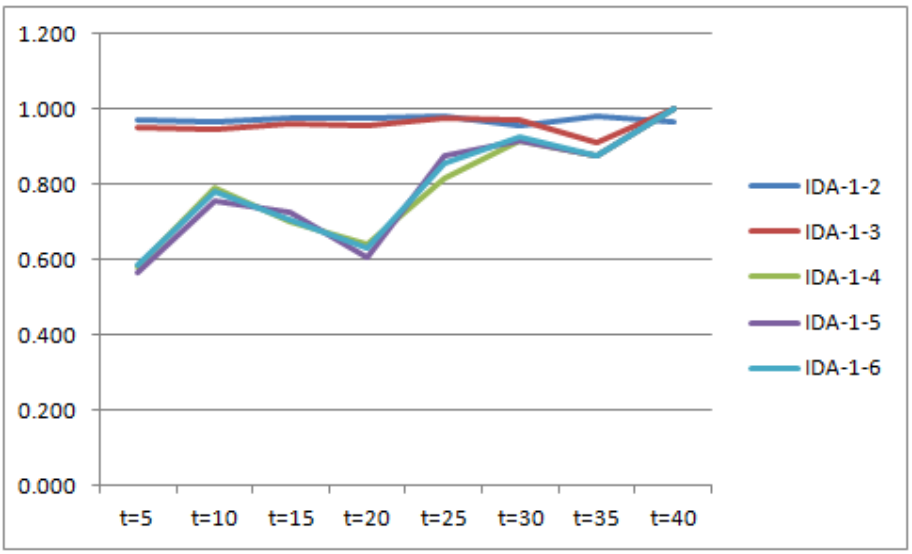


Figure 5.5.: Results of *Repeatability* with varying T in ORB.

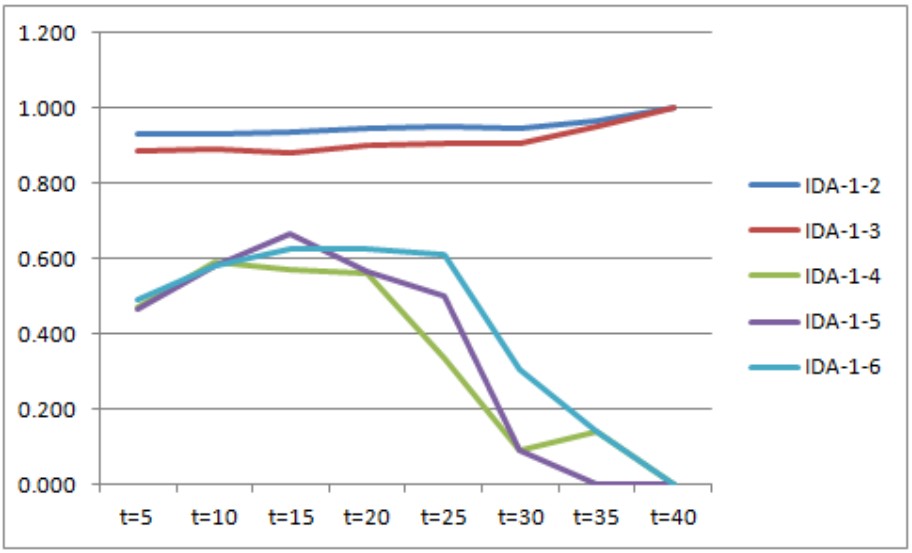


Figure 5.6.: Results of *Recall* with varying T in ORB.

5.3. Cross Check filter

Cross Check filter is introduced in section 3.3.1, this experiment is designed to test, if applying of cross check filter can improve the accuracy of matching result. In this experiment, the *Light* image sequence (see figure 4.4) is used. On each image pairs, number of features and processing are recorded in three situation:

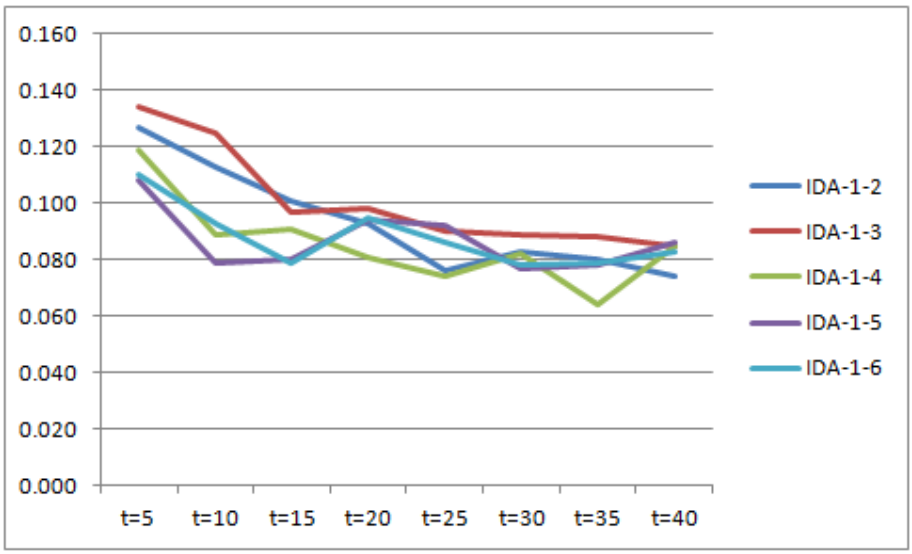


Figure 5.7.: Results of *Duration* with varying T in ORB.

1. Simple matching without filer.
2. Applying cross check filer, for each query feature return only one best matches
3. Applying cross check filter, for each query feature return 3 nearest matches.

The following table 5.9 summarizes the result of this experiment. Comparing the first two columns, after applying the cross check filter, the number of correct matches and the corresponding Recall value are reduced, and the processing time becomes longer. Now comparing the second and third columns, except the slightly difference on Duration metric, all results stay the same.

The cross-check algorithm filters some one-way-matches out, the number of accepted matches is reduced, this affects the RANSAC process. Increasing the number of the returned nearest matches does not change the result. The double matching process increases the duration. The test result demonstrates that, in case of matching two static images, cross check filter does not improve the performance of the whole matching process with RANSAC.

5.4. Time Consumption

Image matching process can be divided into 3 parts: detection, description and matching. In this experiment, the processing time for each part is individually recorded and compared. Four feature descriptors are tested: SIFT, SURF, BRIEF (uses FAST as feature detector) and ORB. This experiment uses the *Boat* sequence (see figure 4.8) from last chapter.

	None Filter					Cross Check (knn=1)					Cross Check (knn=3)				
	N_{qry}	N_{csp}	N_{crt}	R	T	N_{qry}	N_{csp}	N_{crt}	R	T	N_{qry}	N_{csp}	N_{crt}	R	T
SIFT	1770	1154	967	0.838	5.98	1770	1154	930	0.806	6.79	1770	1154	930	0.806	6.78
SURF	2403	2046	1714	0.838	2.30	2403	2046	1448	0.708	2.98	2403	2046	1448	0.708	2.96
BRIEF	7634	6855	6301	0.919	2.04	7634	6855	5413	0.790	4.08	7634	6855	5413	0.790	3.90
ORB	702	527	407	0.772	0.13	702	527	314	0.596	0.14	702	527	314	0.596	0.14
AGAST	2368	2156	1936	0.898	0.39	2368	2156	1734	0.804	0.65	2368	2156	1734	0.804	0.64
BRISK	1001	804	577	0.718	0.15	1001	804	493	0.613	0.20	1001	804	493	0.613	0.20
SU-BRS	1027	808	779	0.964	0.12	1027	808	578	0.715	0.17	1027	808	578	0.715	0.17

Table 5.9.: Results of cross-check-filter experiment. (N_{qry}: number of detected query Keypoints, N_{csp}: number of correspondences, N_{crt}: number of correct matches found after RANSAC), R: Recall value, T: Duration)

The following table 5.10 summarizes the time of each processing step and the figure 5.8 visualizes this result. The result is discussed according to different approaches in details:

- SIFT: the most time-consuming step in SIFT is description, second most time-consuming step is detection. Because of the high computing complexity SIFT takes the most time to extract and describe the features. Matching process is finished in quite short time, the matching algorithms for vector description are efficient.
- SURF: SURF finishes the detection and description in almost equal time. Compared to SIFT, the processing time is significantly reduced. The matching part is also finished in quite short time like SIFT.
- BRIEF: BRIEF consumes quite short time in detection and description, but significantly much more time on matching process. FAST is one of the most efficient feature detectors, but too many detected feature points leads to the huge time consumption on matching process, though BRIEF is a binary descriptor, and matching of binary description is easier and faster than vector description.
- ORB: all three steps in ORB require almost the same time and are finished in quite short time. Compared to other descriptors tested in this experiment, ORB performs much better than other three. The speed advantage of using binary description is clearly demonstrated in this case.

Duration	SIFT			SURF			BRIEF			ORB		
	det	dsp	mat	det	dsp	mat	det	dsp	mat	det	dsp	mat
RZ2-1-2	1.81	3.308	0.076	0.748	0.796	0.062	0.046	0.141	7.987	0.062	0.047	0.047
RZ2-1-3	1.669	3.151	0.062	0.718	0.733	0.062	0.031	0.125	6.677	0.063	0.031	0.062
RZ2-1-4	1.638	3.03	0.046	0.702	0.686	0.047	0.031	0.109	5.553	0.062	0.032	0.062
RZ2-1-5	1.607	2.956	0.062	0.686	0.655	0.032	0.031	0.109	4.633	0.078	0.032	0.046
RZ2-1-6	1.528	2.793	0.062	0.671	0.655	0.047	0.031	0.110	3.728	0.062	0.031	0.047

Table 5.10.: Results of time consumption experiment. (det: detection, dsp: description, mat: matching, unit: s)

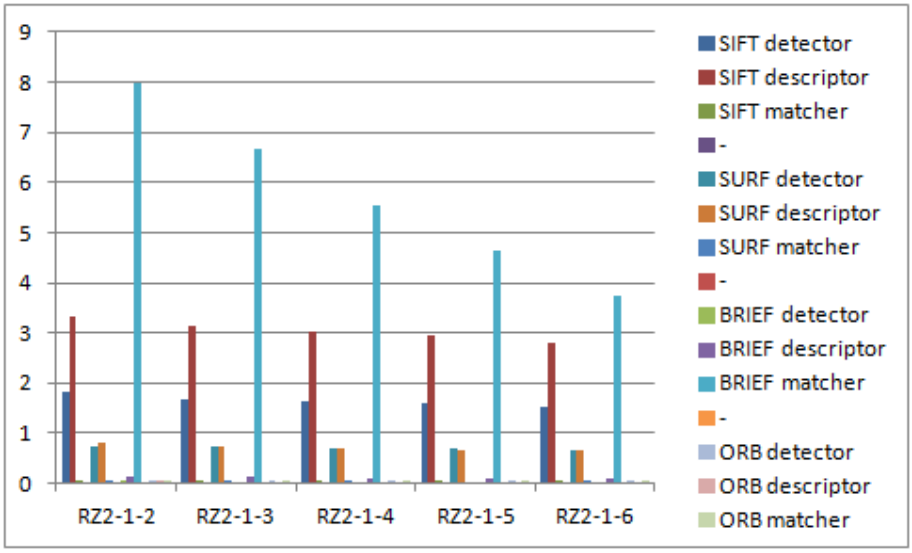


Figure 5.8.: Results of time consumption experiment.

6. Real-time experimentation

After several experiments on static image pairs for comparing the performance of different feature descriptors, this chapter shows the experimental results on a real-time application.

6.1. Implementation

The *Integrating Vision Toolkit* (IVT) [1] is a platform-independent open source C++ computer vision library with an object-oriented architecture. It offers a clean camera interface and a general camera model, as well as many fast implementations of image processing routines and mathematic data structures and functions, such like SIFT and Harris-SIFT are avaiable in IVT. The IVT is compatible with OpenCV, In fact it integrates part of OpenCV functions by optional wrappers, a class called "IplImageAdaptor" realizes the image format conversion between the IVT and OpenCV. Since ORB and BRISK show outstanding performances in the previous experiments in the chapter 4, it is reasonable to test them with real-time application. The ORB implementation from OpenCV and BRISK implementation from its developer based on OpenCV [2] are used in this experiment.

6.2. GUI

The IVT offers also its own multi-platform GUI toolkit, which is used for the implementation of this real-time experiment. The figure 6.1 shows the main window. It consists of two panels - control panel and display panel. In the right part of the control panel there is a combo box to choose the type of feature descriptor, 4 feature descriptors are available here: SIFT, Harris-SIFT, ORB and BRISK. Three parameters can be set in the left part of the control panel. The type of parameters changes according to the type of feature descriptor. For instance, by SIFT the value of "quality threshold", "matching threshold" and "kd-tree leaves" can be adjusted (see figure 6.1) and when ORB is chosen, the parameters become "t", "number of feature" and "knn in cross check" (see figure 6.2). The user can also decide, whether RANSAC applies at the end of the image matching process or not. The matching results is displayed on the under panel in real-time. The left part of the display panel shows the live streaming captured by camera or video sequences, the right part show the reference frame, which is

[1] http://ivt.sourceforge.net/
[2] http://www.asl.ethz.ch/people/lestefan/personal/BRISK

chosen manually by pressing the button "space". The duration and the number of found matches are also displayed in GUI.

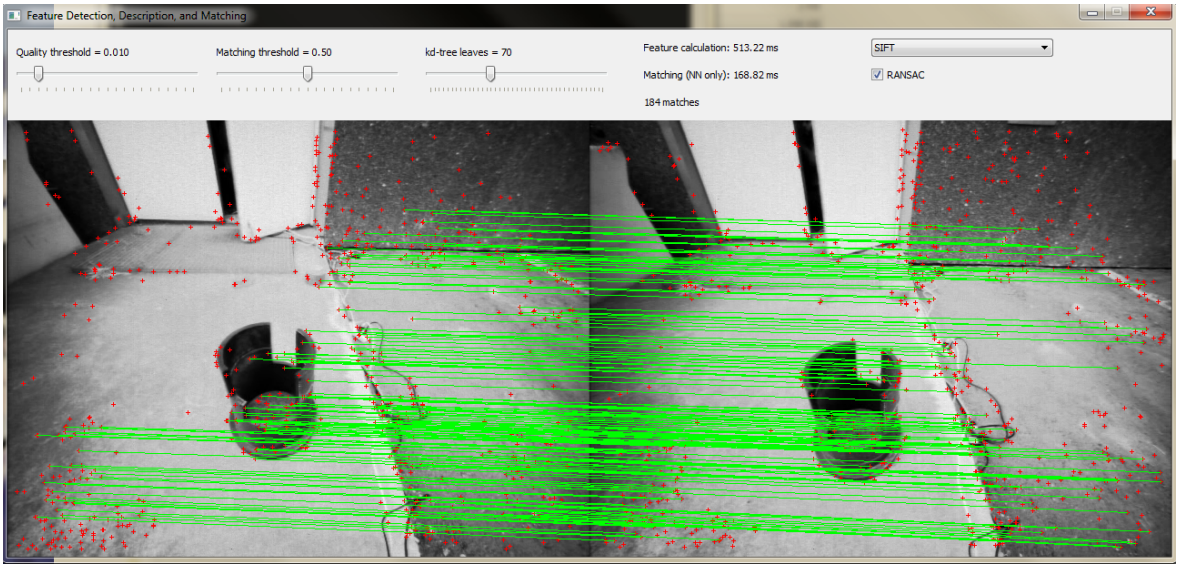


Figure 6.1.: GUI of the experimental real-time application

6.3. Experimental methodology

At the earlier experiment phase, the application uses the live streaming captured by a camera as the input. The whole experiment requires one or two people to manipulate, one holds the camera in hand and moves around, while the other one sets the descriptor and reference image manually and observes the matching results. In order to enhance the repeatability of experimental results, the application has been improved to read a pre-recorded video as input optionally. But the setting of descriptor type, corresponding parameters and the reference frame remains manually. The videos are taken in both indoor and outdoor environments. The motion of camera is intended to contain rotation, scale and viewpoints changes.

6.4. Results

6.4.1. Descriptor comparison

The application processes the video sequences frame to frame, only when the whole image matching process is finished on current frame, the next frame is then read and processed. This mechanism causes a delayed display, when the entire matching requires longer time than the frame rate of the video sequence. Display lag is particularly evident on SIFT.

Although the Harris-SIFT has optimized some steps of SIFT already, the lag is reduced, it is still unable to achieve the requirements of real-time application. The processing speed of ORB and BRISK is quite fast, there is no display lag observed by the naked eyes, the video sequence plays smoothly.

In matching results aspect, SIFT and Harris-SIFT shows outstanding performance, a considerable number of feature keypoints are detected, and the matching result is accurate and robust. Particularly after applying RANSAC, the displayed matching pairs are always correct. Although the ORB and BRISK detect less feature keypoints comparing to SIFT, they also get the sufficient number of correct matches.

In conclusion, despite the high accuracy and robustness, SIFT and Harris-SIFT are not suitable for a real-time application because of the long processing time. ORB and BRISK achieve the real-time requirements and shows a good performance meanwhile.

6.4.2. Cross check filter

During the experiment, an interesting phenomenon is observed: when the number of detected feature matches falls down to a threshold, many query feature points may match to one single point in the reference frame (see figure 6.2). They are obviously wrong matches. In order to prevent such mismatching, a cross check filter is added. The algorithm of cross check filter is introduced in section 3.3.1. In the GUI, the size of K is allowed to select by user.

"K = 0" means none filter applied, sample matching. The one-to-many miss matching occurs sometimes;

"K = 1" represents the using of a cross check filter, and for each query feature only the best match is returned. After swapping the image pair and matching the features from reference image to the query image, do the cross check, only when the both features are the best match to each other, this matches is accepted. With help of cross check filter, the one-to-many mismatching disappears.

"K = 2, 3, ..." indicates that for one query feature point two or more nearest matches are returned for cross check. Based to observation, with increasing K the number of successful matches is slightly reduced, and the processing time is slightly increased. When K > 5, the matching results and processing time are no longer changed.

6.4.3. Logarithmic camera

The advantage of the logarithmic camera is the less sensitivity to illumination changes. This advantage is not evident for the slowly sunshine change in outdoor environments, but in indoor environments with fluorescent lamp, the difference between two adjacent frames captured by ordinary camera and logarithmic camera are quite obvious. Since the previous experiment in section 4.5.1 has demonstrated that SIFT, ORB and BRISK features

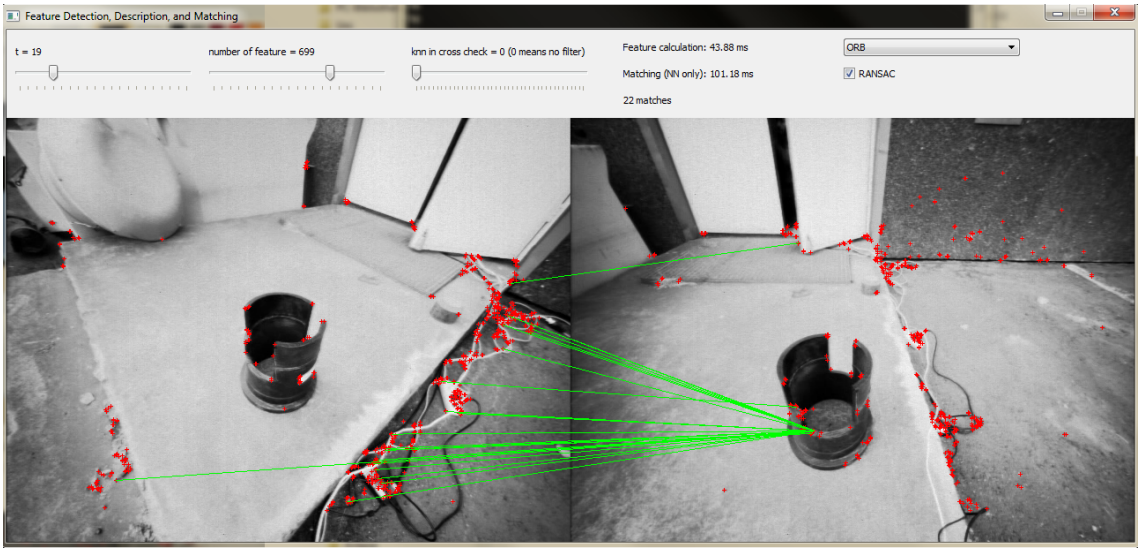


Figure 6.2.: One-To-Many mismatching in real-time application

show outstanding performance under substantial illumination changes, in this case, it is not necessary to use logarithmic camera in a real-time application.

7. Conclusions and Future Work

This thesis represents an introduction and comparison of the most popular feature matching methods. Three experimental evaluations are done for the comparison purpose.

A performance comparison of different feature descriptors is implemented first. The test image sequence contain the most common image deformation: illumination changes, blur, rotation and zoom, viewpoint change and image compression. The results are analyzed in seven performance metrics: number of Keypoints, Repeatability, Recall, Efficiency, Duration, Speed and average Distance. Considering all comparison result form chapter 4, ORB and BRISK are the best choice for a mobile VO-System. They both perform well under all previously mentioned condition of changes. If it is expected in advance that the image sequence has no rotation and scale deformation at all, ORB and SU-BRISK are suggested. In case of a fixed camera position, BRIEF and SU-BRISK are the first choice.

Then the focus is shifted to improvements of the image matching process. Three experiments are designed for this purpose: using lens with different focal length and saving image with different bit depth, adjusting the parameters and applying a cross check filter. The test result has demonstrated: saving the image in more bit depth does not help to improve the performance for the whole image matching process; using lens with larger focal length, the performance under strong illumination has no change, the performance under weak illumination is significantly enhanced; in ORB matching process, T=10 and T=15 are more suitable for the *4-2mm* sequence than the default parameter value T=20. But considering all measures, T=20 is a appropriate default value for all kind of images; in case of matching two static images, cross check filter does not improve the performance of the whole matching process with RANSAC. After that, another experiment is executed to analyze the time consumption of each step in image matching process.

The last experimental evaluation is intended for the real-time application. Despite the high accuracy and robustness, SIFT and Harris-SIFT are not suitable for a real-time application because of the long processing time. ORB and BRISK achieve the real-time requirements and show a good performance meanwhile. When the number of detected feature matches falls down to a threshold, many query feature points may match to one single point in the reference frame. Applying an cross check filter can prevent this one-to-many mismatching. Logarithmic camera shows no advantage in a real-time application.

Future Work Because of the limitations of the time, there are some unimplemented ideas and spaces for improvement.

In the comparison program, all feature descriptors use the default parameter setting, no adjustment according to the different images and scenes. Section 5.2 has confirmed that adjusting the parameter T can improve the performance of the ORB features. When all feature descriptors are adjusted to their optimal performance, the comparison result is more reasonable and convincing. One directions for the future research could be automatically adjusting the parameters based on image analysis to achieve the optimal matching result.

In recent years the research field of image matching has rapidly developed, there may be some new published feature descriptors which have better performance but are not mentioned In this thesis.

All matching methods discussed in this thesis are applied on whole image. In order to maintain a more uniform distribution of feature points, the image can be cut into small segments first, then feature detector is applied on each segment. This idea has been mentioned in [SF11]. It can ensure that each region of the image contains a sufficient number of feature points. This applies especially to feature detector producing a limited limited number of keypoints, such like ORB, which sorts all features based on Harris corner measure and then return only the top 700 features. In particular scenario it may occur that all 700 points are concentrated in one small area of the image, the advantage of segmentation is obvious in this case.

For the real-time application, although the experimental repeatability has been enhanced by reading the pre-recorded video sequences instead of live streaming captured by a camera as the input, all settings like type of descriptor, value of parameters even the capturing of reference image are still done manually, it is still impossible to repeat two exactly identical experiments. If all the manual setting can be automated, the experiment will be stricter.

The matching result of real-time experiment is presented in form of images and real-time numeric display. All conclusions in section 6.4 were obtained based on observation through human eyes. It is better to record all data into a file or database first then post-analyze by computer later. And right now only the number of matches and processing time are displayed on GUI, more performance matrices and analysis results can be added.

The reference frame should captured automatically. In a real VO-application, the reference frame may be changed according to the time or under particular conditions. For instance, update the reference frame in every 5 seconds, or replace the old reference frame once the number of feature matches falls below a threshold. Instead of matching the features extracted from reference frame, matching the best features from all previous frames are more reasonable. A database to store the best feature so far and their descriptions is necessary.

A. Installation

A.1. CMake

As the build system, CMake has to be installed for the first place. It can be downloaded from the CMake-homepage [1] .

A.2. OpenCV

1. Download OpenCV-Library from OpenCV-homepage [2].
2. Install it correctly. (see Install Guide here: [3])
3. The ORB implementation is included in OpenCV since version 2.3.0, make sure the installed version is newer then 2.3.0. The Comparison Program uses OpenCV 2.3.1., if the other version of OpenCV is installed, open the CMake-Configure-File "FindOpenCV.cmake" (under the folder "/config") and replace all number "231" with corresponding version number.
4. Make sure the path to OpenCV-lib-folder, OpenCV-bin-folder and OpenCV-include-folder are recorded in system environment.

A.3. CMake Configuration

1. Start CMake-GUI.
2. At the entry "Source Code", choose the root-path (not "/src" !) of the Descriptor Comparison project.
3. At the entry "Binaries", type the path "{Root-Path}/build".
4. Click the button "Configure", choose the specific IDE and compiler, then click the button "Finish".

[1] http://www.cmake.org/cmake/resources/software.html
[2] http://opencv.willowgarage.com/wiki/
[3] http://opencv.willowgarage.com/wiki/InstallGuide

5. Now check the echo message, if all OpenCV libraries are found correctly (ignore the warning about "pthread"). If one or more libraries can not be found, either type the path of the libraries manually, or check the system path. (After changing the system path, CMake-GUI has to be restarted to load the new system path).
6. If the path of all OpenCV libraries are shown correctly, click the button 'Configure' again, then the button "Generate".
7. After the Project-file is generated, close the CMake-GUI.

A.4. Compiling

1. Open the Comparison Program with project-file "{Root-Path}/build/Descriptor_Comparison.∗ " (The ending ∗ is verify depending on the different IDE which is selected in the last CMake configuration step.)
2. Compile the project.
3. Start the binary program under the path "{Root-Path}/build/src". Note that the OpenCV libraries are dynamic, if the path to "OpenCV-Bin-Folder" is not added into the system path, all corresponding ∗.dll files are required to be copied into the same folder as the execution program.

Bibliography

[AAD09] P. Azad, T. Asfour, R. Dillmann. Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition. In *IROS*, pp. 4275–4280. IEEE, 2009. (Zitiert auf den Seiten 10 und 18)

[ABH+10] M. Achtelik, A. Bachrach, R. He, S. Prentice, N. Roy. Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-denied Indoor Environments. In *Proceedings of SPIE*, p. 7332. SPIE, 2010. (Zitiert auf Seite 13)

[BETG08] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359, 2008. (Zitiert auf den Seiten 10 und 19)

[BL97] J. S. Beis, D. G. Lowe. Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pp. 1000–. IEEE Computer Society, Washington, DC, USA, 1997. URL `http://dl.acm.org/citation.cfm?id=794189.794431`. (Zitiert auf Seite 24)

[Cal10] M. Calonder. *Robust, High-Speed Interest Point Matching for Real-Time Applications*. Ph.D. thesis, Computer Vision Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2010. (Zitiert auf den Seiten 5, 21, 22, 29 und 30)

[CG08] A. Cumani, A. Guiducci. Fast stereo-based visual odometry for rover navigation. *WSEAS Trans. Cir. and Sys.*, 7(7):648–657, 2008. (Zitiert auf Seite 13)

[CLSF10] M. Calonder, V. Lepetit, C. Strecha, P. Fua. BRIEF: Binary Robust Independent Elementary Features. In K. Daniilidis, P. Maragos, N. Paragios, editors, *ECCV (4)*, volume 6314 of *Lecture Notes in Computer Science*, pp. 778–792. Springer, 2010. (Zitiert auf den Seiten 10 und 29)

[CMR10] A. I. Comport, E. Malis, P. Rives. Real-time Quadrifocal Visual Odometry. *I. J. Robotic Res.*, 29(2-3):245–266, 2010. (Zitiert auf Seite 13)

[Cor04] D. S. S. Corke, P.; Strelow. Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pp. 4007–4012. 2004. (Zitiert auf Seite 13)

[FA91] W. T. Freeman, E. H. Adelson. The Design and Use of Steerable Filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, 1991. (Zitiert auf Seite 11)

[FB81] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. (Zitiert auf Seite 24)

[FBF77] J. H. Friedman, J. L. Bentley, R. A. Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977. (Zitiert auf Seite 23)

[GBG10] A. C. G. Babbar, P. Bajaj, M. Gogna. Comparative study of image matching algorithms. *International Journal of Information Technology and Knowledge Management*, 2:337–339, 2010. (Zitiert auf Seite 9)

[GMU96] L. J. V. Gool, T. Moons, D. Ungureanu. Affine/ Photometric Invariants for Planar Intensity Patterns. In *Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I*, ECCV '96, pp. 642–651. Springer-Verlag, London, UK, UK, 1996. (Zitiert auf Seite 11)

[KD87] J. J. Koenderink, A. J. van Doom. Representation of local geometry in the visual system. *Biol. Cybern.*, 55(6):367–375, 1987. (Zitiert auf Seite 11)

[Kleo8] B. Kleiner. *Generierung von Stützinformationen aus optischen Systemen für inertiale Navigationssysteme*. Master's thesis, Technische Universität Dresden, 2008. (Zitiert auf Seite 13)

[KM07] G. Klein, D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*. Nara, Japan, 2007. (Zitiert auf Seite 20)

[Kru13] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. *Sitzungsberichte der Mathematisch Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften*, 122:1939–1948, 1913. (Zitiert auf Seite 14)

[KS04] Y. Ke, R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *CVPR (2)*, pp. 506–513. 2004. (Zitiert auf den Seiten 10, 18 und 29)

[LCS11] S. Leutenegger, M. Chli, R. Siegwart. BRISK: Binary Robust invariant scalable keypoints. In D. N. Metaxas, L. Quan, A. Sanfeliu, L. J. V. Gool, editors, *ICCV*, pp. 2548–2555. IEEE, 2011. (Zitiert auf den Seiten 5, 10, 23 und 29)

[LH87] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In M. A. Fischler, O. Firschein, editors, *Readings in computer vision: issues, problems, principles, and paradigms*, chapter A computer algorithm for reconstructing a scene from two projections, pp. 61–62. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. (Zitiert auf Seite 14)

[LJo9] O. G. Luo Juan. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing*, 3:143–152, 2009. (Zitiert auf den Seiten 11 und 29)

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision, 20–25 September, 1999, Kerkyra, Corfu, Greece, Proceedings*, volume 2, pp. 1150–1157. 1999. (Zitiert auf den Seiten 10 und 17)

[Low04] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. (Zitiert auf den Seiten 5, 10, 17, 19 und 23)

[LWZ11] Y. R. Lijun Wei, Cindy Cappelle, F. Zann. GPS and Stereovision-Based Visual Odometry: Application to Urban Scene Mapping and Intelligent Vehicle Localization. In *International Journal of Vehicular Technology*, volume 2011, p. 17. 2011. doi:10.1155/2011/439074. (Zitiert auf Seite 13)

[Mat89] L. H. Matthies. *Dynamic stereo vision*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1989. AAI9023429. (Zitiert auf Seite 13)

[MCM07] M. W. Maimone, Y. Cheng, L. Matthies. Two years of Visual Odometry on the Mars Exploration Rovers. *J. Field Robotics*, 24(3):169–186, 2007. (Zitiert auf Seite 13)

[MHB+10] E. Mair, G. D. Hager, D. Burschka, M. Suppa, G. Hirzinger. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In K. Daniilidis, P. Maragos, N. Paragios, editors, *ECCV (2)*, volume 6312 of *Lecture Notes in Computer Science*, pp. 183–196. Springer, 2010. (Zitiert auf den Seiten 10 und 20)

[Mor80] H. P. Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Ph.D. thesis, Stanford University, Stanford, CA, USA, 1980. AAI8024717. (Zitiert auf Seite 13)

[MS90] L. Matthies, S. A. Shafer. Error modeling in stereo navigation. In I. J. Cox, G. T. Wilfong, editors, *Autonomous robot vehicles*, chapter Error modeling in stereo navigation, pp. 135–144. Springer-Verlag New York, Inc., New York, NY, USA, 1990. (Zitiert auf Seite 13)

[MS05] K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. (Zitiert auf den Seiten 10 und 11)

[Nis03] D. Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. In *CVPR (2)*, pp. 195–202. IEEE Computer Society, 2003. (Zitiert auf Seite 14)

[NNB04] D. Nistér, O. Naroditsky, J. R. Bergen. Visual Odometry. In *CVPR (1)*, pp. 652–659. 2004. (Zitiert auf den Seiten 13, 14 und 25)

[Ols09] E. B. Olson. Real-time correlative scan matching. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 4387–4393. 2009. (Zitiert auf Seite 13)

[OMSM00] C. F. Olson, L. H. Matthies, M. Schoppers, M. W. Maimone. Robust Stereo Ego-motion for Long Distance Navigation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 453–458. 2000. (Zitiert auf Seite 13)

[RD06] E. Rosten, T. Drummond. Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, volume 1, pp. 430–443. 2006. (Zitiert auf den Seiten 5, 20 und 21)

[RPD08] E. Rosten, R. Porter, T. Drummond. Faster and Better: A Machine Learning Approach to Corner Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2008. (Zitiert auf Seite 20)

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, G. R. Bradski. ORB: An efficient alternative to SIFT or SURF. In D. N. Metaxas, L. Quan, A. Sanfeliu, L. J. V. Gool, editors, *ICCV*, pp. 2564–2571. IEEE, 2011. (Zitiert auf den Seiten 10 und 22)

[SCS08] K. A. SAVAN CHHANIYARA, L. D. SENEVIRATNE. VISUAL ODOMETRY TECHNIQUE USING CIRCULAR MARKER IDENTIFICATION FOR MOTION PARAMETER ESTIMATION. In *Proceedings of the Eleventh International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines. Advances In Mobile Robotics*, 1069-1076. 2008. (Zitiert auf Seite 13)

[SF11] D. Scaramuzza, F. Fraundorfer. Visual Odometry [Tutorial]. *IEEE Robot. Automat. Mag.*, 18(4):80–92, 2011. (Zitiert auf den Seiten 13 und 80)

[SKK10] A. Schmidt, M. Kraft, A. J. Kasinski. An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation. In L. Bolc, R. Tadeusiewicz, L. J. Chmielewski, K. W. Wojciechowski, editors, *ICCVG (2)*, volume 6375 of *Lecture Notes in Computer Science*, pp. 251–259. Springer, 2010. (Zitiert auf Seite 11)

[SLC99] A. M. S. Lacroix, R. Chatila. Rover self localization in planetary-like environments. In *5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp. 433–440. Noordwijk (The Netherlands), 1999. (Zitiert auf Seite 13)

[SS08] D. Scaramuzza, R. Siegwart. Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *IEEE Transactions on Robotics*, 24(5):1015–1026, 2008. (Zitiert auf Seite 13)

[SZ02] F. Schaffalitzky, A. Zisserman. Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?". In A. Heyden, G. Sparr, M. Nielsen, P. Johansen, editors, *ECCV (1)*, volume 2350 of *Lecture Notes in Computer Science*, pp. 414–431. Springer, 2002. (Zitiert auf Seite 11)

[TRD09] S. Taylor, E. Rosten, T. Drummond. Robust feature matching in 2.3μs. In *IEEE CVPR Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond*. 2009. URL `http://mi.eng.cam.ac.uk/~sjt59/papers/taylor_2009_robust.pdf`. (Zitiert auf Seite 20)

Declaration

All the work contained within this thesis, except where otherwise acknowledged, was solely the effort of the author. At no stage was any collaboration entered into with any other party.

(Zhen Peng)